WHEN RELIABILITY MATTERS

MDC-MAX

Basic Setup Guide

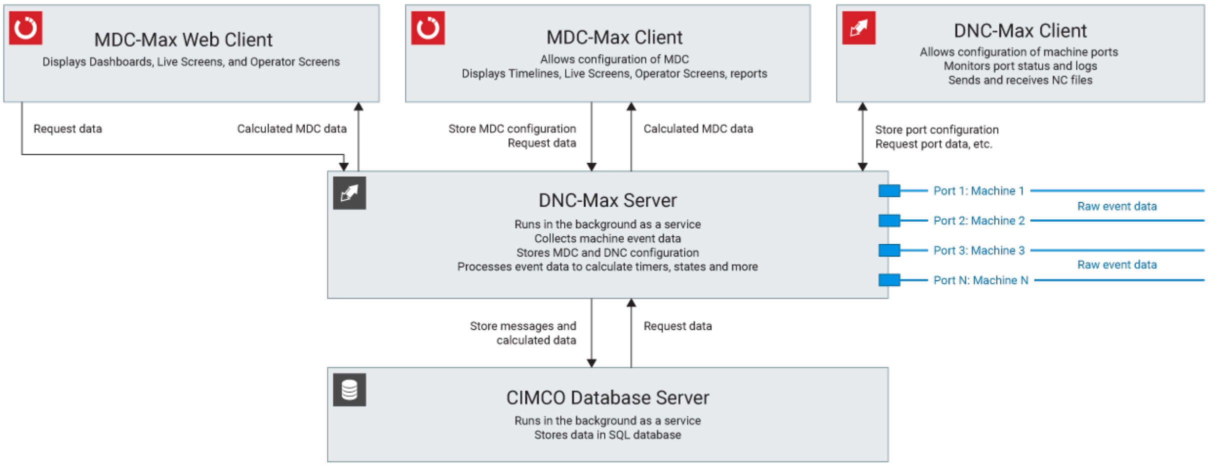
1. Introduction

CIMCO MDC-Max is a software product that enables companies to monitor their machine tools in real-time. It does this by generating messages each time an event occurs (such as the machine starting or stopping) and stores these messages in its database. The MDC-Max Client is then used to generate real-time status screens and historical reports of machine activity from this database.

The goal of this guide is to provide you with a working level of information required to install and configure the components of the MDC-Max system and get an understanding of how they work together. You will learn how to configure machine connections, collect messages from machines and display the collected data in both a Timeline and a Live Screen in the MDC-Max Client.

Once you have installations established your knowledge will expand rapidly and many of the features described in this guide will become second nature to you. Until then, you can use this guide as a reference to help you answer many common configuration questions.

1.1. Components of MDC-Max


MDC-Max Web Client
Displays Dashboards, Live Screens, and Operator Screens
MDC-Max Client
Allows configuration of MDC
Displays Timelines, Live Screens, Operator Screens, reports
DNC-Max Client
Allows configuration of machine ports
Monitors port status and logs
Sends and receives NC files
Request data
Calculated MDC data
Store MDC configuration
Request data
Calculated MDC data
Store port configuration
Request port data, etc.
DNC-Max Server
Runs in the background as a service
Collects machine event data
Stores MDC and DNC configuration
Processes event data to calculate timers, states and more
Port 1: Machine 1
Port 2: Machine 2
Port 3: Machine 3
Port N: Machine N
Raw event data
Raw event data
Store messages and calculated data
Request data
CIMCO Database Server
Runs in the background as a service
Stores data in SQL database

First of all, MDC-Max is not a stand-alone product. It is a client application that works "on top" of the DNC-Max Server, which again works on the CIMCO Database Server. The MDC-Max Client is used to configure the MDC system and to display Timelines, Live Screens, Operator Screens, reports and so forth. The actual configuration is not stored in the client itself, but on the DNC-Max Server/database.

The MDC-Max Web Client is a browser-based client that can display Dashbords, Live Screens, Operator Screens and Shop Floor Screens. The web client works on any device that has a browser available such as tablets, phones, PCs and TVs - the web client, however, cannot be used to configure the MDC system.

The DNC-Max Server is the core component of the MDC system. It handles communication with machine tools and is the central data collection source for all MDC messages. The DNC-Max Server also takes care of processing the message-data it receives and calculates timers, states and so forth. The server must be running at all times and you cannot run any of the clients without it.

Whenever a string of text or line change is detected by the DNC-Max Server (on any of its ports), it converts these to an MDC message. For example, when using a serial connection the CTS high signal could be converted to a CYCLESTART message to indicate that the machine is in cycle (running). When the CTS line goes low this could be converted to a CYCLESTOP message.

The DNC-Max Server is linked to the CIMCO Database Server. That way, all messages, configurations, etc. are automatically stored in a defined database on the CIMCO Database Server. Storage of messages is, of course, essential since otherwise the DNC-Max Server and the MDC-Max client would not have any historic data to work with.

In the diagram, you will also notice the DNC-Max Client. This is used to configure the machine ports on the DNC-Max Server and to configure how each port converts the signal data it receives into MDC messages. The DNC-Max Client is also very useful in regards to viewing both server and port logs which is essential when configuring, testing, and troubleshooting the MDC system.

1.2. Naming

For customers, understanding the relationship between the MDC-Max Client, DNC-Max Server and CIMCO Database Server can be overwhelming. To keep it simple, our MDC solution is simply referred to as MDC-Max and presented as if it was one product. In our marketing material, the DNC-Max Server and CIMCO Database Server are sometimes also referred to as the MDC-Max Server and MDC-Max Database. Again, this is only to make it less confusing and to present it as a single solution.

Although the server is called DNC-Max Server, it is not limited to working with the DNC-Max Client. Think of the DNC-Max Server as a multipurpose server that works at the core of CIMCO's MDC and DNC systems.

You might also notice that the filenames associated with some of the software does not match the software name such as the filename for the CIMCO Database Server which is called NCBaseServer.exe (and located in the /NCBase/ folder) or the DNC-Max Client which is called DNCAdmin.exe. These files have simply retained their original names while the software itself has evolved heavily.

1.3. Before we begin

For this guide, we assume you have already downloaded the latest version of CIMCO Software from **www.cimco.com/download/** and installed the following applications as a minimum:

- MDC-Max Client
- DNC-Max Client
- DNC-Max Server
- Database Server

For this guide, application files should be installed to the directory **C:\CIMCO\** and you should be running version 8 or later.

Expect both the download and installation to take some time.

2. Install NC-Base Database Server

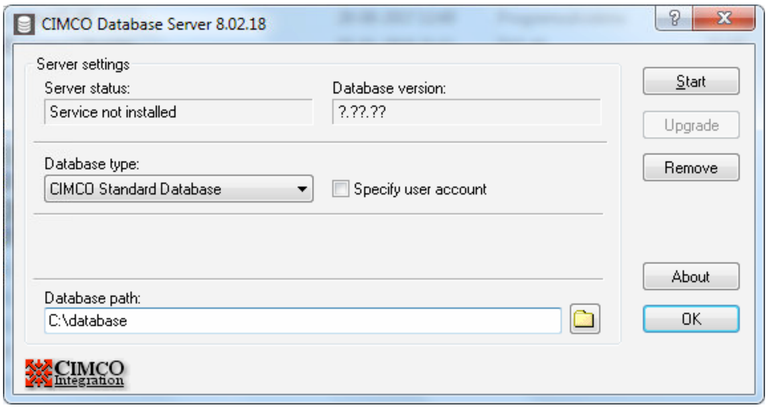
The NC-Base Database Server runs in the background as a Windows service. Its primary purpose is to manage one or multiple SQL databases and to store, process and retrieve data from these databases. The database server also has a huge array of functions for handling versioning, backup, import and so forth.

For this guide, however, all we need is to create a database.

1 First, log on to Windows using an account with full administrative privileges.

2 To open the CIMCO Database Server console use the **Start menu** and go to **All apps** (All programs) > **CIMCO V8** > **Database Server 8**.

The exe file can also be located in **C:\CIMCO\NCBase8\NCBaseServer.exe**.



CIMCO Database Server 8.02.18

Server settings

Server status: Service not installed

Database version: ?.??.??

Database type: CIMCO Standard Database

□ Specify user account

Database path: C:\database

3 For this guide, we will use the standard database. Select the **CIMCO Standard Database** from the **Database type** dropdown.

4 In the field **Database path**, set the database path to **C:\database**. For this guide we want to keep the database files outside of the directory C:\CIMCO. That way, the database files do not get mixed in with the CIMCO software application files and can easily be removed afterwards if needed.

For production environments, however, we recommend keeping the database files in their default location within the C:/CIMCO/ folder. The files are thereby somewhat hidden and will not as easily get deleted by accident. A customer's IT department might want the database files located somewhere else. For more information about this topic please refer to the CIMCO Database Server documentation.

5 Since no database has been created yet, the field **Database version** will display **?.??.??**. To create the CIMCO Standard Database click the **Start** button.

Please note: If the harddisk is being heavily used by some process, this can delay the response from the database service which might result in a time-out error.

6 Once the database has been created, a dialog will appear asking to install the sample database. For this guide we want a clean database without any sample/test data. Click the **No** button. The database version number will now be shown in the field **Database version** which also indicates that the database is properly set up and running.

7 Close the database server console. Remember, the database server still runs in the background as a service and will automatically start if the PC is rebooted.

3. Install and configure the DNC-Max Server

There are two ways to run the DNC-Max Server. Either using the DNC-Max Server application or using the DNC-Max Service Manager application that enables you to install the server as a Windows service.

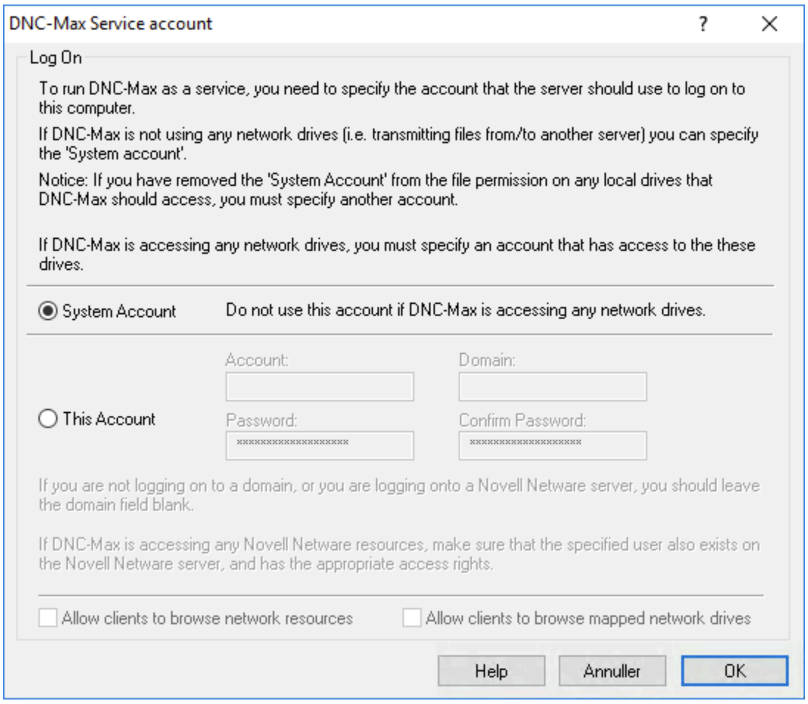
In this guide, we will use the DNC-Max Service Manager to install the DNC-Max Server as a Windows service that runs in the background and automatically starts if the PC is rebooted.

Please note: You will need a valid license key to run the DNC-Max Server as a service. If you try to start the server in trial mode you will get an error stating "Failure accessing the install service".

3.1. Installation

1 To install the DNC-Max server as a service, navigate to **C:\CIMCO\DNCMax8** and double-click the file **DNCMaxServiceManager.exe**. This will open the CIMCO DNC-Max Service Manager dialog.

2 Click the install button to install the server as a service. You will be asked what Windows account the DNC-Max service should use to log on. For this guide you can simply use the **System Account**. Click **OK** to continue.



Log On

To run DNC-Max as a service, you need to specify the account that the server should use to log on to this computer.

If DNC-Max is not using any network drives (i.e. transmitting files from/to another server) you can specify the 'System account'.

Notice: If you have removed the 'System Account' from the file permission on any local drives that DNC-Max should access, you must specify another account.

If DNC-Max is accessing any network drives, you must specify an account that has access to the these drives.

● System Account Do not use this account if DNC-Max is accessing any network drives.

○ This Account

Account:

Domain:

Password: ********************

Confirm Password: ********************

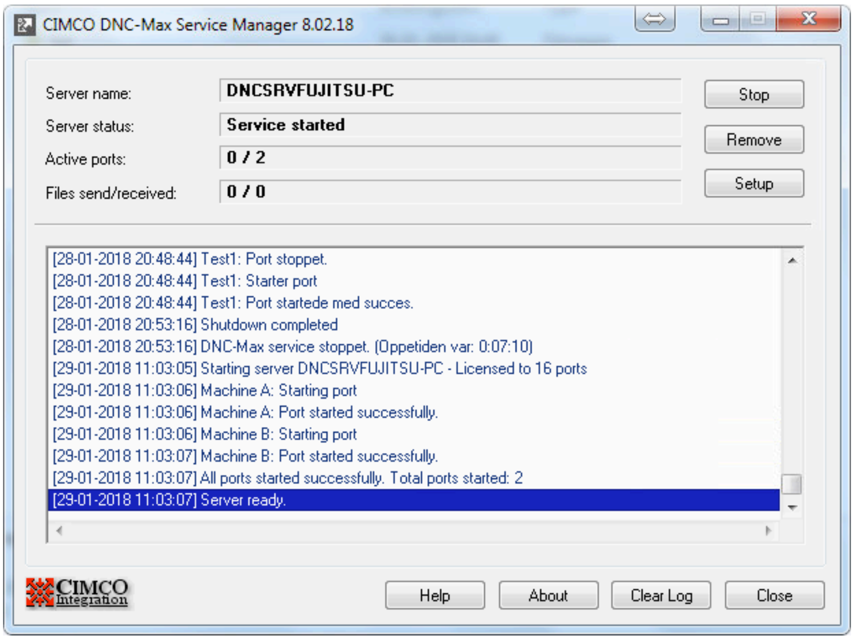
If you are not logging on to a domain, or you are logging onto a Novell Netware server, you should leave the domain field blank.

If DNC-Max is accessing any Novell Netware resources, make sure that the specified user also exists on the Novell Netware server, and has the appropriate access rights.

□ Allow clients to browse network resources □ Allow clients to browse mapped network drives

3 To start the server, click the **Start** button. The DNC-Max Server will now start and run as a service in the background. The service will automatically start every time the PC is rebooted.

4 Once the server is started, you will see values in the fields **Active ports** and **Files send/receive** indicating that the server is running properly.



CIMCO DNC-Max Service Manager 8.02.18

Server name: DNCSRVFUJITSU-PC

Server status: Service started

Active ports: 0 / 2

Files send/received: 0 / 0

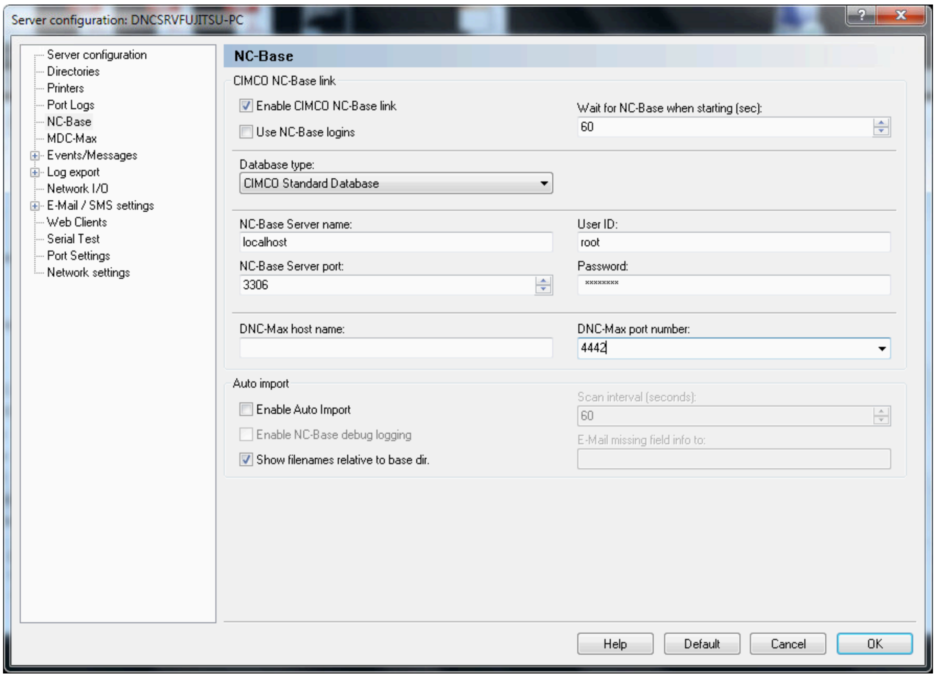
```
[28-01-2018 20:48:44] Test1: Port stoppet.
[28-01-2018 20:48:44] Test1: Starter port
[28-01-2018 20:48:44] Test1: Port startede med succes.
[28-01-2018 20:53:16] Shutdown completed
[28-01-2018 20:53:16] DNC-Max service stoppet. (Oppetiden var: 0:07:10)
[29-01-2018 11:03:05] Starting server DNCSRVFUJITSU-PC - Licensed to 16 ports
[29-01-2018 11:03:06] Machine A: Starting port
[29-01-2018 11:03:06] Machine A: Port started successfully.
[29-01-2018 11:03:06] Machine B: Starting port
[29-01-2018 11:03:07] Machine B: Port started successfully.
[29-01-2018 11:03:07] All ports started successfully. Total ports started: 2
[29-01-2018 11:03:07] Server ready.
```

3.2. Link-up database and MDC-Max

To enable the DNC-Max Server to work with both the database and the MDC-Max Client, we need to enable its NC-Base and MDC-Max links.

1 In the **DNC-Max Service Manager dialog**, click the **Setup** button to enter the **Server configuration dialog**.

2 In the treeview menu, select **NC-Base** and check the option **Enable CIMCO NC-Base link**. After doing this, the fields below become active.



NC-Base

CIMCO NC-Base link

☑ Enable CIMCO NC-Base link

☐ Use NC-Base logins

Wait for NC-Base when starting (sec): 60

Database type: CIMCO Standard Database

NC-Base Server name: localhost

User ID: root

NC-Base Server port: 3306

Password: ********

DNC-Max host name:

DNC-Max port number: 4442

Auto import

☐ Enable Auto Import

☐ Enable NC-Base debug logging

☑ Show filenames relative to base dir.

Scan interval (seconds): 60

E-Mail missing field info to:

Enabling the NC-Base link establishes a connection between the DNC-Max Server and the NC-Base Database Server. This also "tells" the DNC-Max Server to start storing MDC messages, configurations, and more in our database. In the following we specify which NC-Base Server and specific database to use.

3 From the dropdown menu **Database type**, select the **CIMCO Standard Database**.

4 Keep the default **NC-Base Server name** (localhost) and the default **NC-Base Server port** (3306). Enter "**root**" as the database **User ID** and enter the default **Password** "**cimco123**".

The field **DNC-Max host name** can be left blank. In rare cases where clients experience connection problems, entering the host name of the PC where the DNC-Max Server is running can help. Do not specify the IP address of the DNC-Max Server if your network is using DHCP. For more information, see the documentation.

5 Make sure the **DNC-Max port number** is set to the default value **4442**.

Please note: The DNC-Max port number defines the port used by clients to communicate with the DNC-Max Server. This is not limited to the DNC-Max client, since both the MDC-Max Client, NC-Base Client and DNC-Max Client all use this port.

6 Finally, in the treeview menu, select **MDC-Max** and check the option **Enable MDC-Max link**. Leave the other fields checked as well.

7 Click the **OK** button to save the changes. When asked, confirm restart of server and wait a moment for the server to restart. The DNC-Max Server always need to restart when changes are made to its configuration.

3.3. Configure MDC messages

All the functionality in MDC-Max is based on messages (also referred to as events), but you might ask - where do these messages come from? Actually, it is the DNC-Max Server that creates all messages. As we will see later, it is the machine ports on the server that defines when and what kind of message should be created.

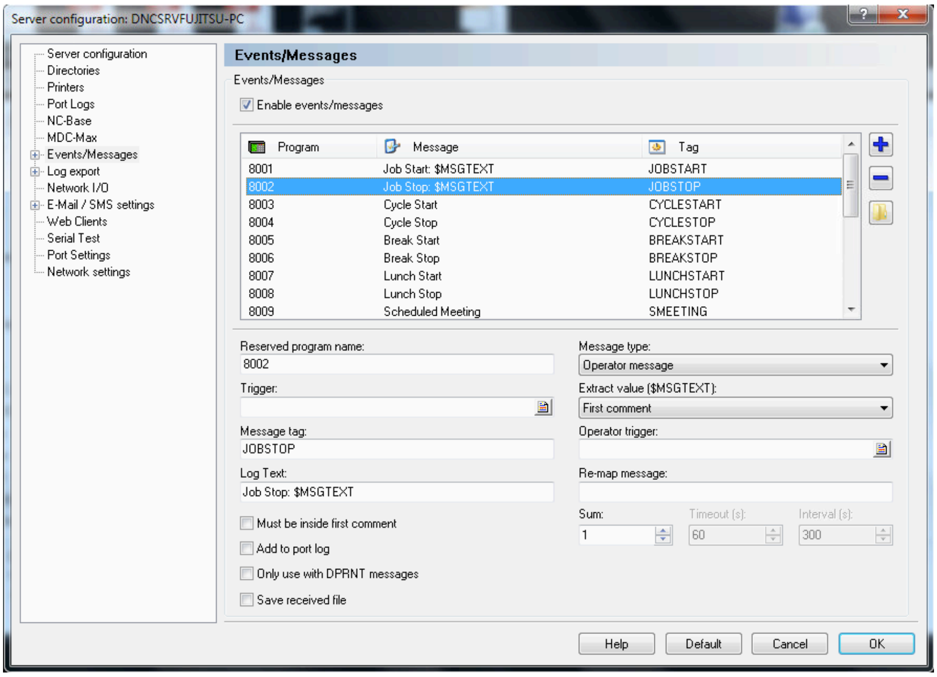
For example, when a CTS high signal is detected on a serial machine port, this port can be configured to create (or trigger) a CYCLESTART message. When the CTS low signal is detected on the port, this can create a CYCLESTOP message. We are not talking about physical hardware ports, but about the ports defined and configured on the DNC-Max Server. These virtual ports receive the raw data from the network or cables connected to the server PC and, based on how the ports are configured, the ports (DNC-Max Server) create messages.

Messages can also be created by Operator Screens when buttons are pushed or input received. We will return to this later in this guide. For now, we will focus on configuring the messages that will be created by our virtual machine ports. We will need the messages CYCLESTART, CYCLESTOP, ALARMON, and ALARMOFF.

1 In the **DNC-Max Service Manager dialog**, click the **Setup** button to (again) enter the **Server configuration dialog**.

2 Select **Event/Messages** in the treeview menu and check the field **Enable events/messages** at the top.

As you can see, there is already a long list of predefined messages. For this guide, we want to use the messages CYCLESTART and CYCLESTOP to determine if our machine is running or not and the messages ALARMON and ALARMOFF to determine if the machine is down. The first two are already set up by default, but we need to manually add the messages **ALARMON** and **ALARMOFF**.



Events/Messages

Events/Messages

☑ Enable events/messages

Program	Message	Tag
8001	Job Start: $MSGTEXT	JOBSTART
8002	Job Stop: $MSGTEXT	JOBSTOP
8003	Cycle Start	CYCLESTART
8004	Cycle Stop	CYCLESTOP
8005	Break Start	BREAKSTART
8006	Break Stop	BREAKSTOP
8007	Lunch Start	LUNCHSTART
8008	Lunch Stop	LUNCHSTOP
8009	Scheduled Meeting	SMEETING

Reserved program name: 8002

Message type: Operator message

Trigger:

Extract value ($MSGTEXT): First comment

Message tag: JOBSTOP

Operator trigger:

Log Text: Job Stop: $MSGTEXT

Re-map message:

Sum: 1 Timeout (s): 60 Interval (s): 300

□ Must be inside first comment

□ Add to port log

□ Only use with DPRNT messages

□ Save received file

3 To add a new message you can either duplicate an existing message and modify the duplicate or you can create a new message. To create a duplicate, select the message you want to duplicate and click the [+] icon on the right side of the list. To create a new message scroll to the bottom of the list and click outside on the empty row so no message is selected - then click the [+] icon. We will use the duplicate method.

Select the message **Cycle Start** and click the [+] icon to duplicate it. Enter **ALARMON** in the **Message tag** field and **Alarm On** in the **Log Text** field. Delete the existing value in the field **Reserved Program Name** since this is not needed.

The **Message tag** is the unique identifier for the message and is always written in UPPERCASE, without spaces, and only using English characters (no local characters like Ø, Â, Ü, etc.).

The **Log Text** is the human readable name of the message which is used in log entries.

4 We also need to create the **ALARMOFF** message. Duplicate the **ALARMON** message, we just created, and change the **Message tag** to **ALARMOFF** and the **Text Log** to **Alarm Off**.

5 When done, click the **OK** bottom at the bottom of the dialog and click **Yes** when asked to restart the server. The server must restart in order to activate the changes.

Once restarted, verify that the line **Running MDC multi-threaded engine...** is shown in the server log. That way you know MDC is activated.

6 Click the **Close** button, to close the DNC-Max Service Manager.

Please note: It is possible to test the MDC setup (and the messages) without connecting an actual machine. This can be achieved by creating an Operator Screen in the MDC-Max Client with buttons for triggering the different machine messages. We will show how this is done later in this guide after we have created and configured a machine port on the DNC-Max Server.

4. Data collection using MOXA Nport 5110

The first port we will configure on the DNC-Max Server is one that uses the **Standard Serial Protocol** and a **MOXA Nport 5110**. The **Standard Serial Protocol** is used in situations where the machine does not have a data interface and the hardware signals have to be wired from the machine's PLC relays using a serial cable such as an RS-232 cable. To get the signals from the machine, the serial cable will be connected to a MOXA NPort 5110 device server which is also connected to the LAN using Ethernet (or directly to the test PC).

Please note: For most installations using an Nport 5110 is not recommended. It is only included in this guide to provide a basic starting point and some background knowledge. For serial connections we always recommend using the superior ioLogik E1210 as described later in this guide.

Please refer to the section Connection using MOXA NPort 5110 (simplified) on page 46 for a more detailed description of this type of connection. In the following, we will pretend this is the connection we need to configure and that our customer has already wired the machine in the following way:

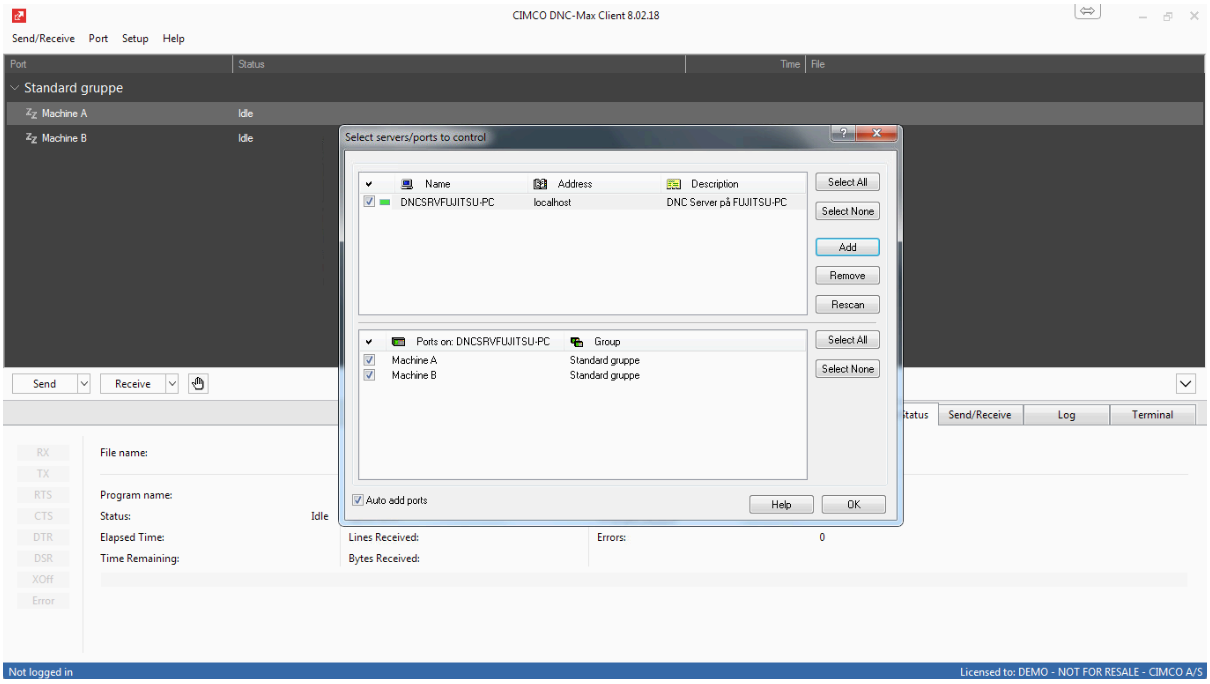
Serial cable pin CTS <-- Machine PLC: Cycle start/stop
Serial cable pin DSR <-- Machine PLC: Alarm on/off

Please note: For this guide, you do not need a Moxa NPort 5110 to complete the configuration.

In order to receive and interpret signals from the machine we need to configure a port on the DNC-Max Server. Ports are configured using the DNC-Max Client.

1 First, open the **DNC-Max Client**. Use the client shortcut already created on the desktop. You will notice that the Port List is empty. This is because the DNC-Max client has not been connected to the DNC-Max Server yet. More specifically, we need to add our DNC-Max Server to the list of servers that the client can control.

2 Add the DNC-Max Server by selecting **Setup / Select Servers** from the menu. This brings up the **Select servers/ports to control** dialog.



✓	Name	Address	Description
☑	DNCSRVFUJITSU-PC	localhost	DNC Server på FUJITSU-PC

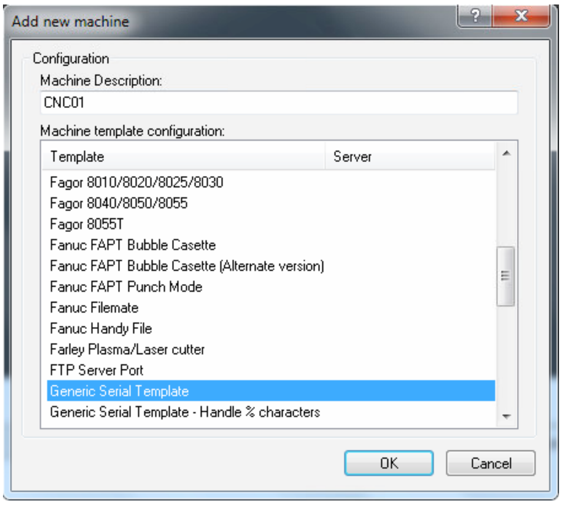
✓	Ports on: DNCSRVFUJITSU-PC	Group
☑	Machine A	Standard gruppe
☑	Machine B	Standard gruppe

3 In the dialog, click the **Add** button and enter **localhost** as the **Server name**. This is the network name of the PC on which the DNC-Max Server is running. In this case, the server is running on the same PC as the DNC-Max Client - that is why we use **localhost**.

4 At the top of the dialog you will now see the DNC-Max Server listed and in the lower part you will see the ports (machines) already configured on the server. The ports **Machine A** and **B** are default ports which we will remove later. Click the **OK** button to close the dialog.

5 As you can see, the ports **Machine A** and **B** are now displayed in the Port list. Ignore these for now.

6 Instead, we need to create the port for our first machine. Select **Port / Add Port** from the top menu. This brings up the **Add new machine** dialog.



Machine Description: CNC01

Machine template configuration:

Template	Server
Fagor 8010/8020/8025/8030	
Fagor 8040/8050/8055	
Fagor 8055T	
Fanuc FAPT Bubble Casette	
Fanuc FAPT Bubble Casette (Alternate version)	
Fanuc FAPT Punch Mode	
Fanuc Filemate	
Fanuc Handy File	
Farley Plasma/Laser cutter	
FTP Server Port	
Generic Serial Template	
Generic Serial Template - Handle % characters	

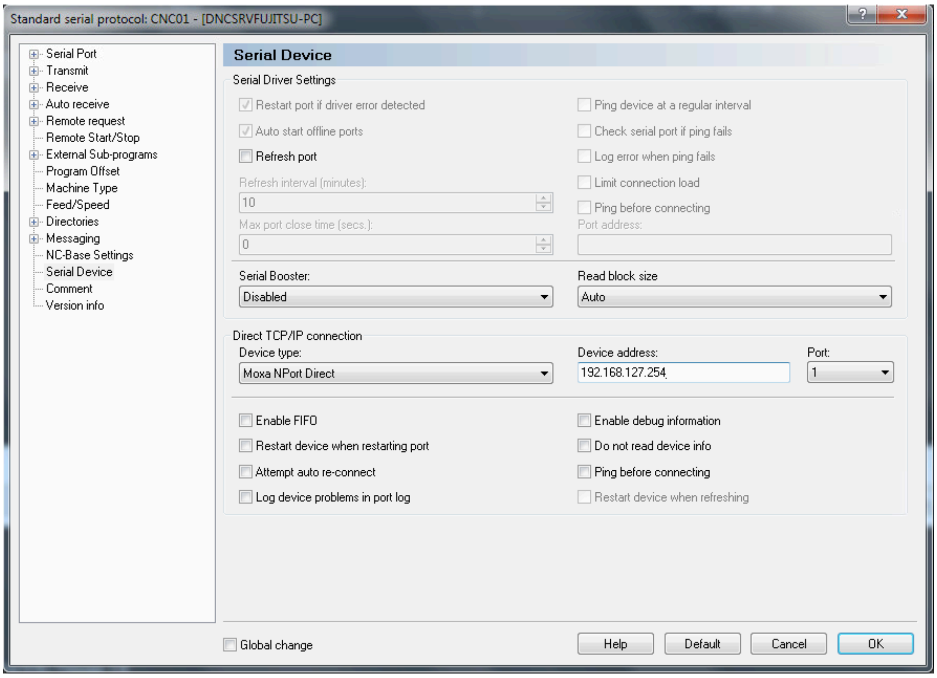
7 In the dialog, enter **CNC01** in the field **Machine Description** or choose a more meaningful name for the machine yourself. In the **Machine template configuration** list below, select **Generic Serial Template** and click **OK** to close the dialog. The port will now be added and the **Standard serial protocol** dialog will appear where we can configure the port.

The port configuration options are determined by what configuration template is used. We chose the **Generic Serial Template** because our physical connection to the machine is using a serial cable for signal transmission.

8 **Please note:** If you do not have a Moxa Nport 5110 connected, you can skip this step.

In the treeview menu, select **Serial Device**. This is where we can configure the type and IP address of the Moxa Nport 5110.

The dropdown **Device Type** should be set to **Moxa Nport Direct** and in the field **Device address** the IP address of the Nport should be entered. Since the Moxa Nport 5110 has one COM port, make sure **Port** is set to use Port **1**.


Standard serial protocol: CNC01 - [DNCSRVFUJITSU-PC]

Serial Device

Serial Driver Settings

☑ Restart port if driver error detected
☑ Auto start offline ports
□ Refresh port
Refresh interval (minutes): 10
Max port close time (secs.): 0

□ Ping device at a regular interval
□ Check serial port if ping fails
□ Log error when ping fails
□ Limit connection load
□ Ping before connecting
Port address:

Serial Booster: Disabled
Read block size: Auto

Direct TCP/IP connection

Device type: Moxa NPort Direct
Device address: 192.168.127.254
Port: 1

□ Enable FIFO
□ Restart device when restarting port
□ Attempt auto re-connect
□ Log device problems in port log

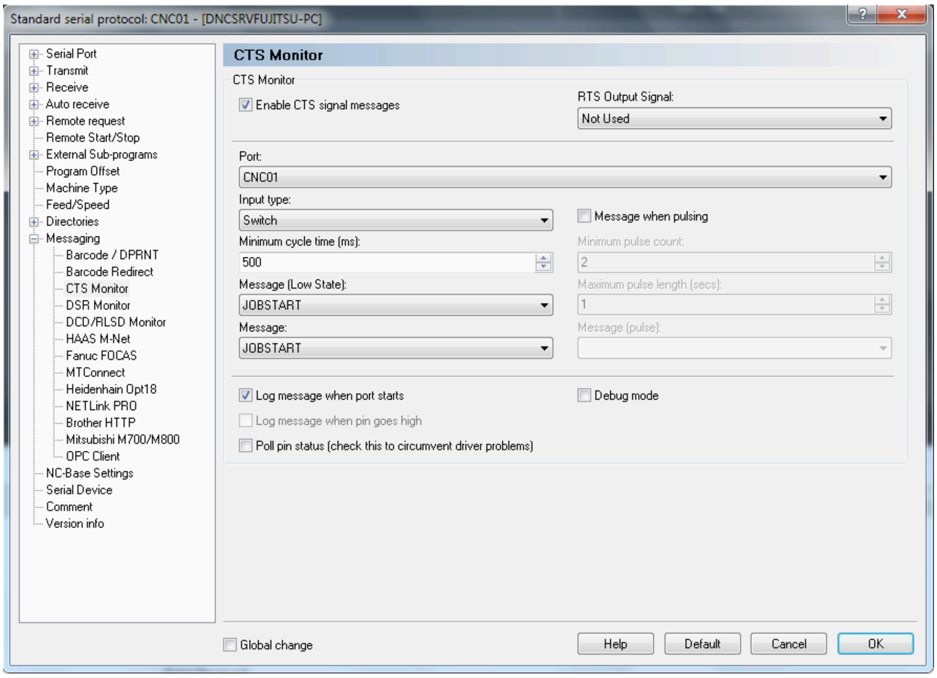
□ Enable debug information
□ Do not read device info
□ Ping before connecting
□ Restart device when refreshing

□ Global change

9 In the treeview menu, select **Messaging** and check the field **Enable events/messages**.

10 In the treeview menu, below **Messaging**, select **CTS Monitor** and check the field **Enable CTS signal messages**.

Earlier we decided that the machine signals for cycle start/stop should be wired to the CTS pin. Now we need to configure what messages should be triggered when the CTS signal changes.



CTS Monitor

CTS Monitor

☑ Enable CTS signal messages

RTS Output Signal: Not Used

Port: CNC01

Input type: Switch

☐ Message when pulsing

Minimum cycle time (ms): 500

Minimum pulse count: 2

Message (Low State): JOBSTART

Maximum pulse length (secs): 1

Message: JOBSTART

Message (pulse):

☑ Log message when port starts

☐ Debug mode

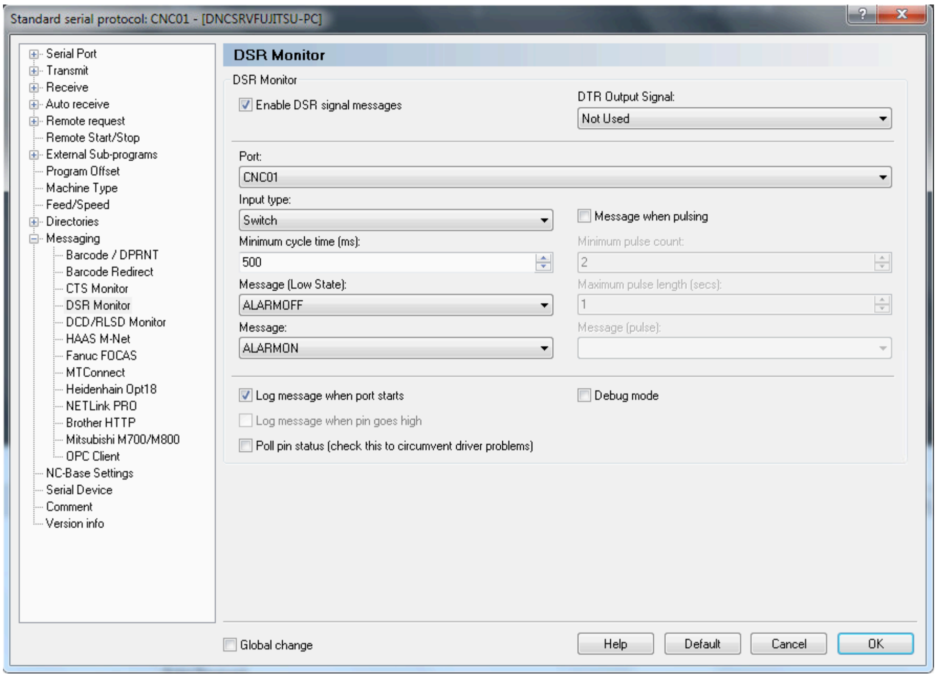
☐ Log message when pin goes high

☐ Poll pin status (check this to circumvent driver problems)

☐ Global change

11 Set the **Input type** to **Switch**. A Switch generates a new message every time the signal changes. The signal can either be in a low or in a high state. Set **Message (Low state)** to trigger the message **CYCLESTOP** and set the high state **Message** to trigger **CYCLESTART**. That way, when the signal goes low it will trigger the CYCLESTOP message and when it goes high it will trigger the CYCLESTART message.

12 Similarly, we need to setup the Alarm On/Off messages on the DSR signal. In the treeview menu, select **DSR Monitor** and check the field **Enable DSR signal messages**. Set **Input type** to use **Switch**, set the **Message (Low state)** to **ALARMOFF** and set the high state **Message** to **ALARMON**.



DSR Monitor

DSR Monitor

☑ Enable DSR signal messages

DTR Output Signal: Not Used

Port: CNC01

Input type: Switch

☐ Message when pulsing

Minimum cycle time (ms): 500

Minimum pulse count: 2

Message (Low State): ALARMOFF

Maximum pulse length (secs): 1

Message: ALARMON

Message (pulse):

☑ Log message when port starts

☐ Debug mode

☐ Log message when pin goes high

☐ Poll pin status (check this to circumvent driver problems)

☐ Global change

13 Click the **OK** button to close the configuration dialog.

Our port is now configured to generate the CYCLESTART/STOP messages based on the CTS signals high/low and the ALARMON/OFF messages based on the DSR signals high/low.

14 Delete the default **Machine A** and **B** ports. Highlight port in list, select **Port / Delete Port** from the top menu.

It is a good idea to enable the **System Log** and **Debug tabs** in the DNC-max Client. This makes it a lot easier to see what is going on and to see the raw data received on the port. System Log shows the DNC-Max Server log and Debug shows the details about the current selected port.

15 Select **Setup / Client Configuration** in the top menu to bring up the **Client configuration** dialog. Check the fields **Show System Log** and **Show Debug**. Click the **OK** button to close the dialog.

5. Create an Operator Screen for testing messages

An easy way to test machine messages is to create an Operator Screen with buttons for triggering the different kinds of messages. That way you can test the different features of MDC-Max without connecting any hardware.

1 Open the **MDC-Max Client** and click the **Login** button in the topmenu. Use the default username **ADMIN** and leave the password field empty. Click the **Login** button.

2 Select the **Setup** tab and click the **System Setup** button. Then click the button **Live Setup**.

3 In the treeview menu, select **Operator Screens** and then click the **Add** button to add a new Operator Screen. In the **Description** field, name it **MDC Test**.

4 To add buttons for CYCLESTART/STOP and ALARMON/OFF, select **Buttons** in the treeview menu.

Notice that the title reads **Buttons: MDC Test**. This is becasue we are currently configuring buttons for the Operator Screen **MDC Test**. If you have multiple Operator Screens, the screen you want to configure must first be selected in the list under **Operator Screens.**

5 Click the **Add** button and in the **Description** field write **Cycle Start**. In the dropdown menu **Push button message** select **CYCLESTART**. Add the remaining buttons in the following way:

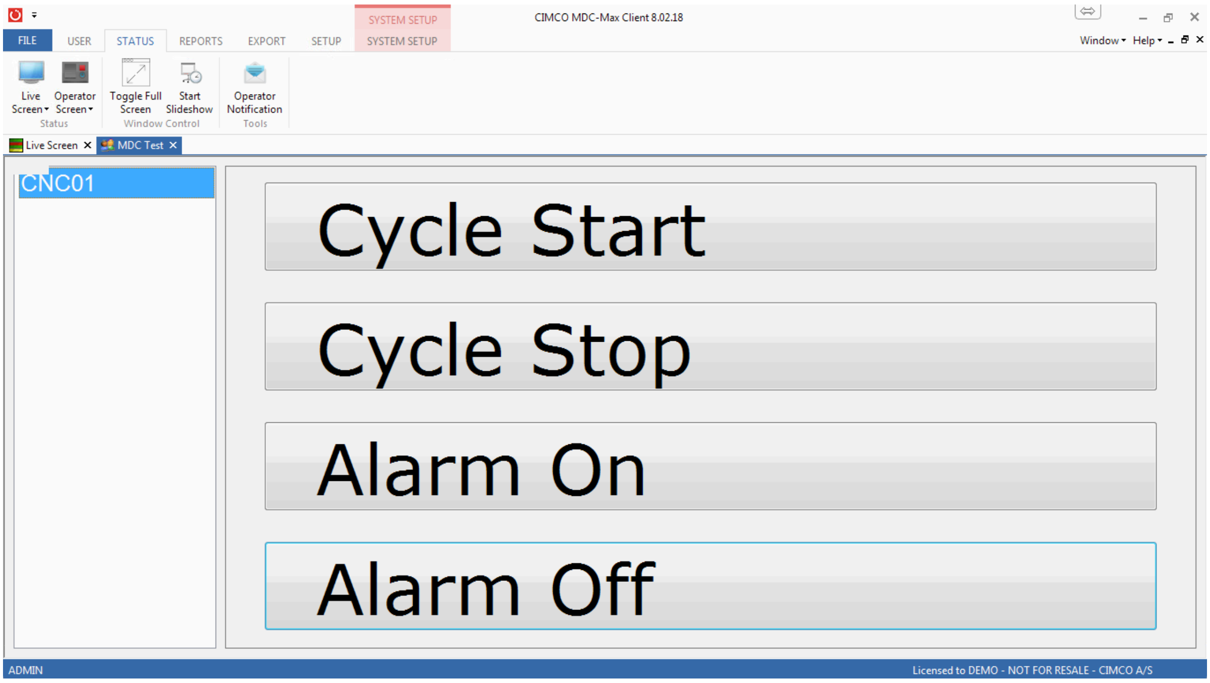
Click Add button. Description: **Cycle Stop**. Push button message: **CYCLESTOP**
Click Add button. Description: **Alarm On**. Push button message: **ALARMON**
Click Add button. Description: **Alarm Off**. Push button message: **ALARMOFF**

Please note: If you do not see any of the above messages in the **Push button message** dropdown menu you need to restart the DNC-Max Server again. But, this time it has to be done through the DNC-Max Client. The DNC-Max Client does some additional "under-the-hood stuff" that makes this work.

To do this, close the MDC-Max Client and Open the DNC-Max Client. From the topmenu select **Setup** > **Restart Server**. Once restarted, open the MDC-Max Client again and return to the Live Setup configuration. Add the buttons to the Operator Screen as described above.

6 When done, click the **Save/Close** button in the top menu. To see the Operator Screen, select the **Status** tab and click the button **Operator Screen**. You should see the buttons you just

created. Click the buttons a couple of times to generate different messages.



7 To verify that the messages are being created, select **Reports** > **Event Log**. In the log, you should see log entries for the messages just created such as CYCLESTART and CYCLESTOP.

6. Create a Timeline and Live Screen

Now that our port and messages for Cycle and Alarm have been configured, it is time to put the messages to use in the MDC-Max Client.

6.1. Create Timers

In order to track for how long our test machine has been in cycle, stopped, or in alarm, we need to create Timers that are controlled by the messages CYCLESTART/STOP and ALARMON/OFF. MDC-Max uses Timers to count the time from the start of one event to another and it is these Timers that generate the information we use in Timelines, Live Screens, and reports.

Please note: By default, Timers are generic for all ports/machines. The benefit of this is that you can easily connect new machines, configure their ports and they will automatically show up in your Timelines, Live Screens, reports, and so forth.

For new ports, you need to make sure that all the messages used by the Timers are also included in the new port configuration. If this is not possible for some reason, the messages that are not configured will naturally not get tracked. This will, however, not break the MDC system.

1. To create Timers, open the **MDC-Max Client**.

2. Select the **Setup** tab and click the **System Setup** button. Then click the button **MDC-Engine**. In the treeview menu, select **Timers**.

3. By default, a timer called **CYCLE** is already created. All we need to do is change its **Start condition** and **Stop condition**, so that the timer is triggered by the **CYCLESTART/STOP**

messages.

Set the message for **Start condition** by clicking the **search icon** to the right of the field. This brings up the **Expression dialog**.

4. Make sure to delete any existing text in the Expression field at the top of the dialog.

   We want the Timer to start when the message **CYCLESTART** is received from the machine. In MDC-Max all messages are denoted by **MSG[nameofmessage]**. Scroll down the list to the section that begins with **MSG**. Click on the **MSG[CYCLESTART]** and click the **Insert** button below so it appears in the **Expression** field. Then click the **OK** button.

5. Similar for the **End condition**. Click the **search icon** to bring up the **Expression dialog**. Find **MSG[CYCLESTOP]** in the list, click **Insert** and then click the **OK** button.

6. For the **Clear Condition**, Click the **search icon** to bring up the **Expression dialog** and scroll to the bottom of the expression list where you will find the **TIME** expressions. Select **TIME[ISDAYEND]** and click the **OK** button.

   MDC-Max has built-in functions that you can use to determine the current time, when a day starts and ends, and other time related values. We want to clear our timer at the end of the day, which by default is set to 18:00. Setting the Clear Condition to TIME[ISDAYEND] means that the current timer (CYCLE in this case) will be reset to 00:00:00 at the end of the day. For more information about how time is configured in MDC-Max, please refer to the section Schedule Setup in the MDC-Max documentation.

7. Our CYCLE Timer only tracks the time when the machine is running, however, we also want to track the time when the machine is not running (when it is stopped). We can achieve this by creating a new Timer that works opposite of the CYCLE Timer, so instead it counts the time from CYCLESTOP to CYCLESTART.

   To add a new Timer, click the **Add** button on the right side of the list of Timers. In the dialog that appears, enter **STOPPED** in the field **New Timer Name** and leave the field **Machine/Group** set to **All**. Click **OK**.

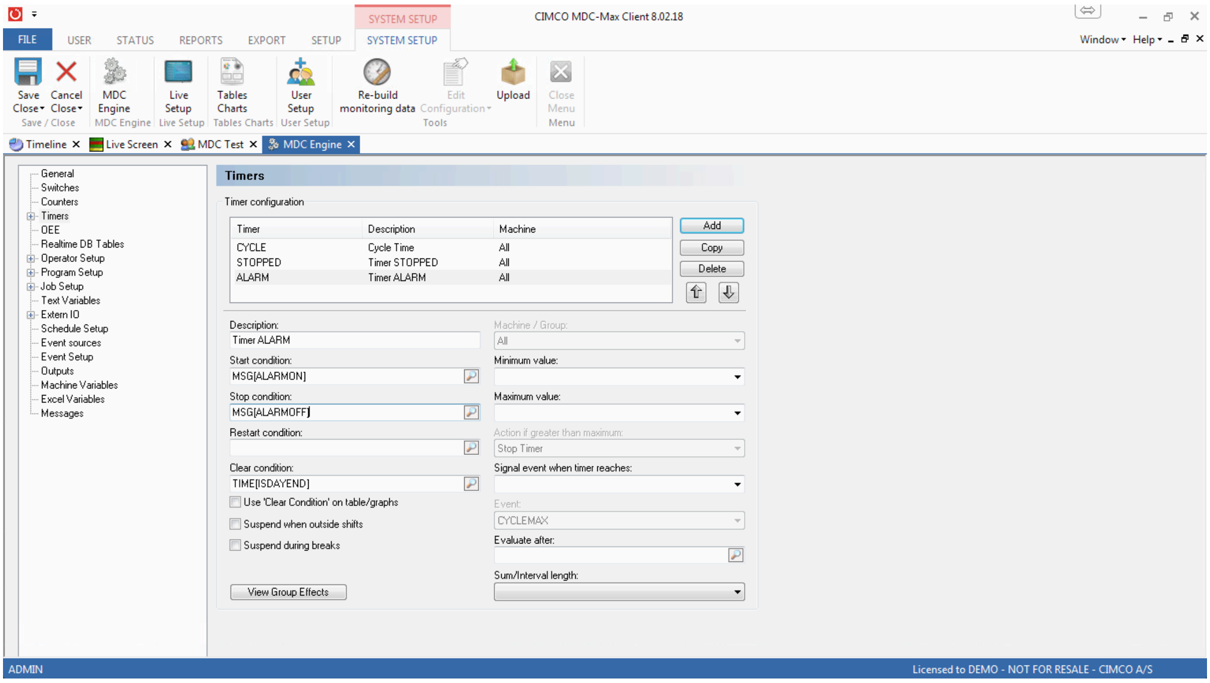
   Set the **Start condition** to **MSG[CYCLESTOP]**, the **End condition** to **MSG[CYCLESTART],** and the **Clear condition** to **TIME[ISDAYSTART]**.

   The Clear condition is set to TIME[ISDAYSTART] because we want the timer to reset at 6:00 and then count stopped time during the day. If we set it to TIME[ISDAYEND] it would clear at 18:00, but then immediately start tracking stopped time. The Stopped timer would then be at 12:00 hours at 6:00 the following day.

   Do not worry about getting this right from the beginning. You can always change the Timer configuration later, and the DNC-Max Server will simply re-calculate the Timers based on all the message data that has been collected.

8. Finally, we need to add an additional Timer for ALARM. Click the **Add** button on the right side of the list of Timers. In the dialog that appears, in the field **New Timer Name** enter **ALARM** and leave the field **Machine/Group** set to **All**. Click **OK**.

   Set the **Start condition** to **MSG[ALARMON]**, the **End condition** to **MSG[ALARMOFF],** and the **Clear condition** to **TIME[ISDAYEND]**.



Timers

Timer configuration

Timer	Description	Machine
CYCLE	Cycle Time	All
STOPPED	Timer STOPPED	All
ALARM	Timer ALARM	All

Description: Timer ALARM

Machine / Group: All

Start condition: MSG[ALARMON]

Minimum value:

Stop condition: MSG[ALARMOFF]

Maximum value:

Restart condition:

Action if greater than maximum: Stop Timer

Clear condition: TIME[ISDAYEND]

Signal event when timer reaches:

□ Use 'Clear Condition' on table/graphs

Event: CYCLEMAX

□ Suspend when outside shifts

Evaluate after:

□ Suspend during breaks

Sum/Interval length:

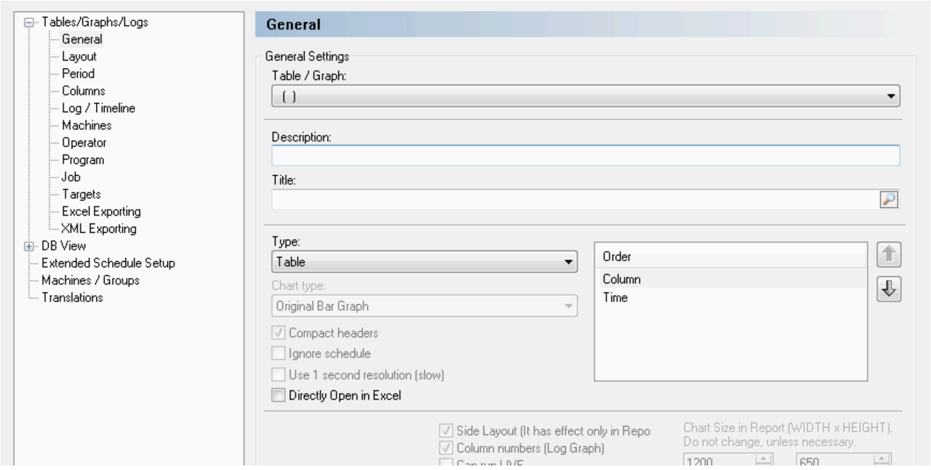
9. Exit the MDC-Engine configuration by clicking the **Save/Close** button in the top menu.

6.2. Create a Timeline

Now that our timers have been configured we can easily create a Timeline to visualize the amount of time our machine has been in cycle, stopped, or in alarm.

1. In MDC-Max, select **Setup** > **System Setup** from the topmenu. Click the button **Tables/Charts** which opens a new tab with the **Tables/Charts configuration**.

2. In the treeview menu, select **Tables/graphs/logs** and click the **Add** button at the buttom of the dialog. Once you do this, you will be sent to the **General** configuration dialog. Notice in the treeview menu that **General** (under **Tables/graphs/logs**) is now selected. This might be confusing at first, but think of it as you just created an empty (undefined) table/graph/log and now you are asked to configure it. This is also indicated in the Table/Graph dropdown menu that has the value ( ).



General

General Settings

Table / Graph: ( )

Description:

Title:

Type: Table

Chart type: Original Bar Graph

Order
Column
Time

☑ Compact headers

□ Ignore schedule

□ Use 1 second resolution (slow)

□ Directly Open in Excel

☑ Side Layout (It has effect only in Repo

☑ Column numbers (Log Graph)

Chart Size in Report (WIDTH x HEIGHT). Do not change, unless necessary.

1200 650

3. In the **Description** and **Title** fields, enter **Timeline**. Also, set **Type** to **Timeline**. So now our Table/Graph is called Timeline and we have defined that it is a Timeline.

4. In the treeview menu, select **Period**. Set **Interval length** to **1 Day**. This is the time interval that the Timeline will display.

5. Since **Port name** is automatically included on Timelines, this step can be skipped.

   To display the port name (machine name) alongside the Timeline, select **Columns** in the treeview menu. Click the **Add** button and in the **Value** field below add the expression **TEXT[PORTNAME]**.

   Once you click the Add button an empty row will be inserted in the list of Columns and it is only once you fill out the Value field that you will see any text in that row.

6. Finally, we have to define what Timers the Timeline should display. In the Timeline we want to display the different time periods in which our Timers have run and to accomplish this we will use the expression TIMERRUNS[nameoftimer].

   In the treeview menu, select **Log/Timeline**. Click the **Add** button and below in the field **Event/Timeline trigger** add the expression **TIMERRUNS[CYCLE]**. In the **Title** field, enter **Cycle** and in the **Color** dropdown menu select **Selected**. Give it a **green** color.

7. Similarly, click the **Add** button and below in the field **Event/Timeline trigger** add the expression **TIMERRUNS[STOPPED]**. In the **Title** field, enter **Stopped** and in the **Color** dropdown menu select **Selected**. Give it a **blue** color.

8. Finally, click the **Add** button again and below in the field **Event/Timeline trigger** add the expression **TIMERRUNS[ALARM]**. In the **Title** field, enter **Alarm** and select a **red** color.

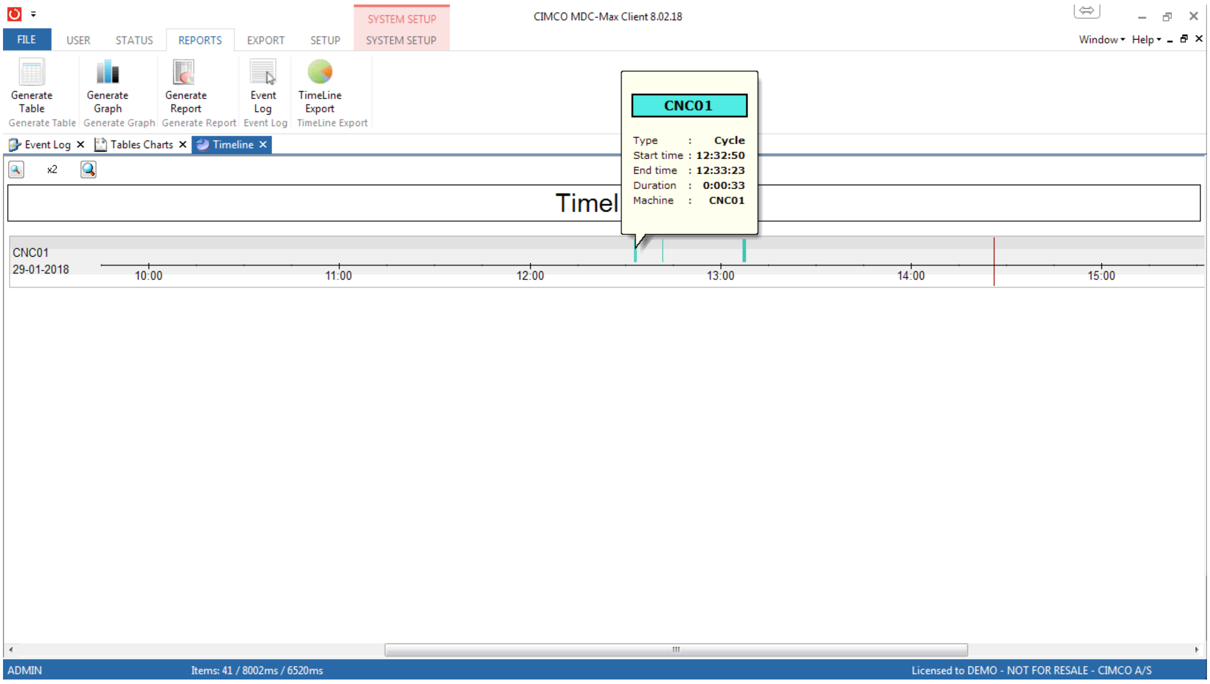
9. Make sure **TIMERRUNS[ALARM]** is moved to the top of the list by using the arrows on the right side of the dialog. This gives the Alarm Timer the highest priority and ensures it will be shown in the Timeline when it is running. If the Alarm Timer had lowest priority the Timeline would only show the Cycle and Stopped Timers since one of these would always be running.

10. Click the **Save/Close** button in the topmenu to exit the configuration.

11. To see the timeline, select **Reports** from the top menu and click the **Generate Graph** button. In the dialog **Select graph to generate**, select the **Timeline** and click the **OK** button to generate the Timeline. You will now see a timeline for our machine CNC01 showing the the current day.

    As the messages from the machine trigger the different Timers, colors will begin to show up in the timeline. If you do not have a machine connected, naturally, no messages will be received. Instead, use the Operator Screen we created to generate the different kinds of messages and see how machine state is reflected in the Timeline.

    Use the zoom icons or the mouse-wheel to zoom all the way in. You will see that the Timeline updates in real-time and status changes are reflected in the Timeline in a matter of seconds.



CNC01

Type : Cycle
Start time : 12:32:50
End time : 12:33:23
Duration : 0:00:33
Machine : CNC01

ADMIN

Items: 41 / 8002ms / 6520ms

Licensed to DEMO - NOT FOR RESALE - CIMCO A/S

6.3. Create a Live Screen

Similar to the Timeline, we will set up a Live Screen to display the current status of our machine in real-time.

1. In MDC-Max, select **Setup** > **System Setup**. Click the button **Live Setup** and select **Status Screens** in the treeview menu.

2. To add a new Live Screen, click the **Add** button. In the **Description** field, enter **Live Screen** as the name of the new Live Screen. Make sure the new Live Screen is selected in the list before moving on.

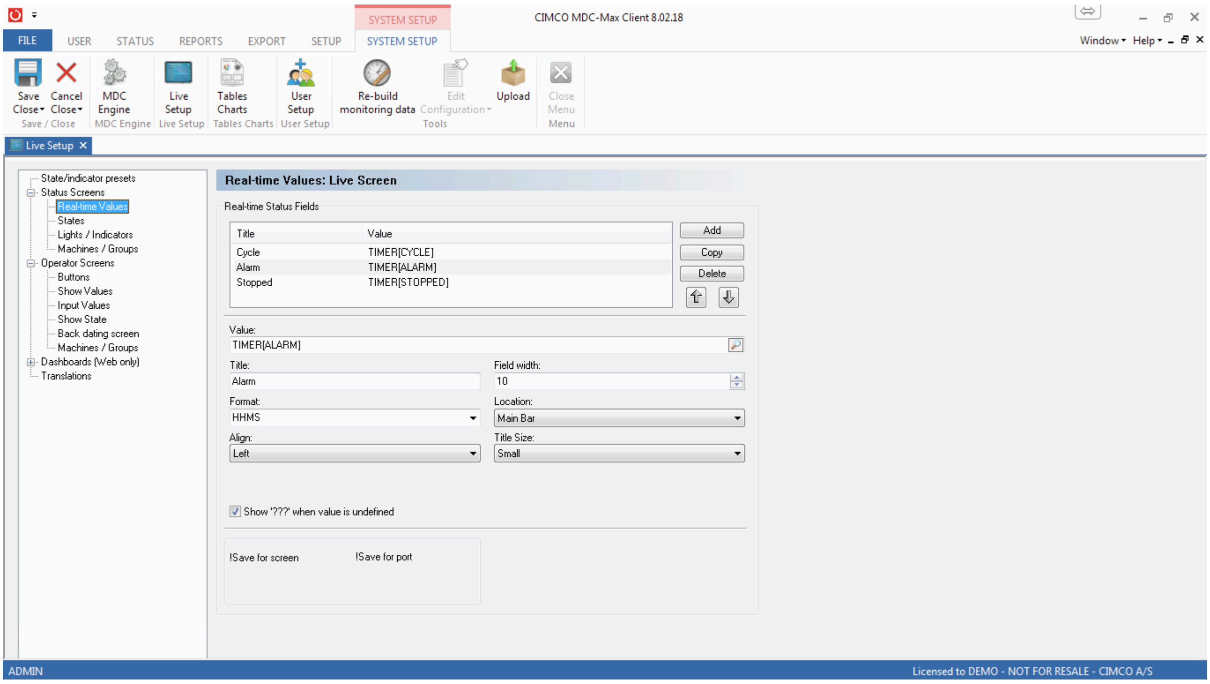
3. In the treeview menu, select **Real-time Values**. This is where we can configure the data that we want to display on our Live Screen. In this case the accumulated time that our machine has been in each of the states cycle, stopped and alarm.

   Click the **Add** button and then click on the **search icon** in the **Value** field. This brings up the Expression dialog. Find and double-click **TIMER[CYCLE]**. Click **OK** when done.

   In the field **Title**, enter **Cycle**. This is the title you want to appear on the Live Screen. Finally, set the **Format** to **HHMS** (hours:minutes:seconds) and **Alignment** to **Left**.

4. In a similar way, add the values for the timers **TIMER[STOPPED]** and **TIMER[ALARM]**.

   Click **Add** button. Value: **TIMER[STOPPED]**. Title: **Stopped**. Format: **HHMS**
   Click **Add** button. Value: **TIMER[ALARM].** Title: **Alarm**. Format: **HHMS**



Real-time Values: Live Screen

Real-time Status Fields

Title	Value
Cycle	TIMER[CYCLE]
Alarm	TIMER[ALARM]
Stopped	TIMER[STOPPED]

Value: TIMER[ALARM]

Title: Alarm

Field width: 10

Format: HHMS

Location: Main Bar

Align: Left

Title Size: Small

☑ Show '???' when value is undefined

!Save for screen

!Save for port

5. Next, we will configure how the Live Screen should look when our machine changes state e.g. green when in cycle and red when in alarm.

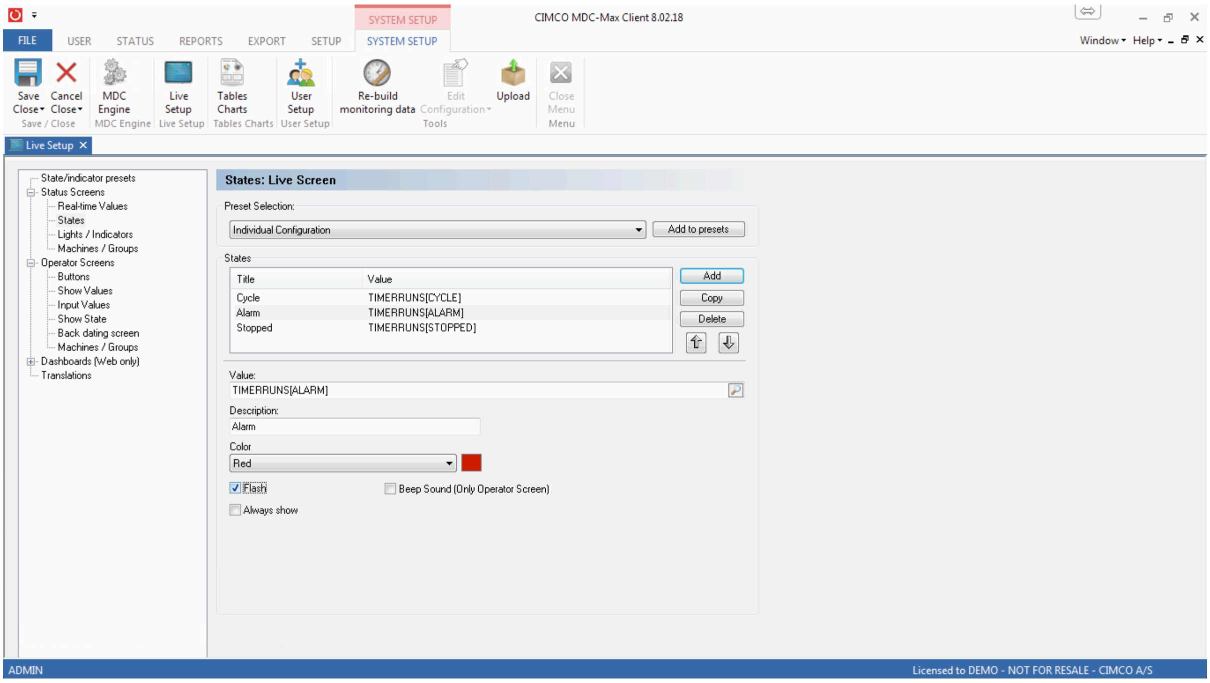
   Select **States** from the treeview menu. Use the **Add** button to add the states **Cycle**, **Stopped,** and **Alarm**. Note that each state is triggered by the **TIMERRUNS[]** expression. So that whenever one of our timers start runnning, the associated state will get triggered. Set up the states in the following way.

   Click **Add**. Value: **TIMERRUNS[CYCLE]**. Description: **Cycle**. Color: **Green**.
   Click **Add**. Value: **TIMERRUNS[STOPPED]**. Description: **Stopped**. Color: **Grey**.
   Click **Add**. Value: **TIMERRUNS[ALARM]**. Description: **Alarm**. Color: **Red**.

   For the Alarm state, enable **Flash** under the **Color** dropdown menu. That way the machine flashes red when in Alarm state.

6. Finally, we want to move the Alarm state on top of the Stopped state, so the order becomes Cycle, Alarm and Stopped. Otherwise the machine will display the stop state even though it stopped and entered the Alarm state. So the placement of States in the list determines their priority.

   In situations where a machine will stop due to some fault, both TIMERRUNS[STOPPED] and TIMERRUNS[ALARM] will get triggered. By having the Alarm state on top of Stopped, the Alarm state will be shown in the Live Screen. Once the problem has been resolved, Alarm Off can be triggered (using the Operator Screen) and the Live Screen will then "fall back" to showing the Stopped state.



States: Live Screen

Preset Selection: Individual Configuration — Add to presets

States

Title	Value
Cycle	TIMERRUNS[CYCLE]
Alarm	TIMERRUNS[ALARM]
Stopped	TIMERRUNS[STOPPED]

Value: TIMERRUNS[ALARM]

Description: Alarm

Color: Red 🟥

☑ Flash

☐ Beep Sound (Only Operator Screen)

☐ Always show

7. Click the button **Save/Close** and select the **Status** tab, then click the **Live Screen** button. The **Live Screen** button only appears if a Live Screen exists.

   Our machine CNC01 will now show in the Live Screen. You can use the Operator Screen to test how the state of the machine changes in the Live Screen when different messages are triggered on the port/machine.

7. Data collection using MOXA ioLogik E1210

We will now add a second machine to our MDC system. We pretend this machine is connected using a MOXA ioLogik E1210 as described in the section Connection using MOXA ioLogik E1210 (simplified) on page 49.

On this machine we have wired the signals Cycle on/off and Alarm on/off so we are capable of receiving the same signals as from the machine CNC01. The machine is wired to the ioLogik in the following way:

ioLogik input (I0) <-- Machine PLC: Cycle start/stop
ioLogik input (I1) <-- Machine PLC: Alarm on/off

We need to create a port for the new machine on the DNC-Max Server and configure this in a way so that the port creates the correct messages based on the machine signals it receives. When setting up a port for a connection using an ioLogik, the procedure is slightly different than with the serial port, since the configuration of messages is not done in the port configuration in the DNC-Max Client, but instead in the DNC-Max Server configuration.

Please note: If you do not have a MOXA ioLogik E1210 you can still follow this guide. The device is not needed to complete the configuration and port messages can still be tested using our Operator Screen.

1. Open the DNC-Max Client and select **Port / Add Port**. In the dialog that appears, enter **CNC02** in the **Description** field and select the **Directory monitor template** in the list. Click **OK** to continue. This brings up the **Network Monitor configuration dialog**. Since we do not need to do anything here, click the **OK** button to close the dialog.

   The **Directory monitor template** is one of the most basic port templates. We chose this because all we really need is to create a simple port to hold the machine name CNC02. The configuration of the signal inputs from the ioLogik and MDC messages is done in the DNC-Max Server configuration.

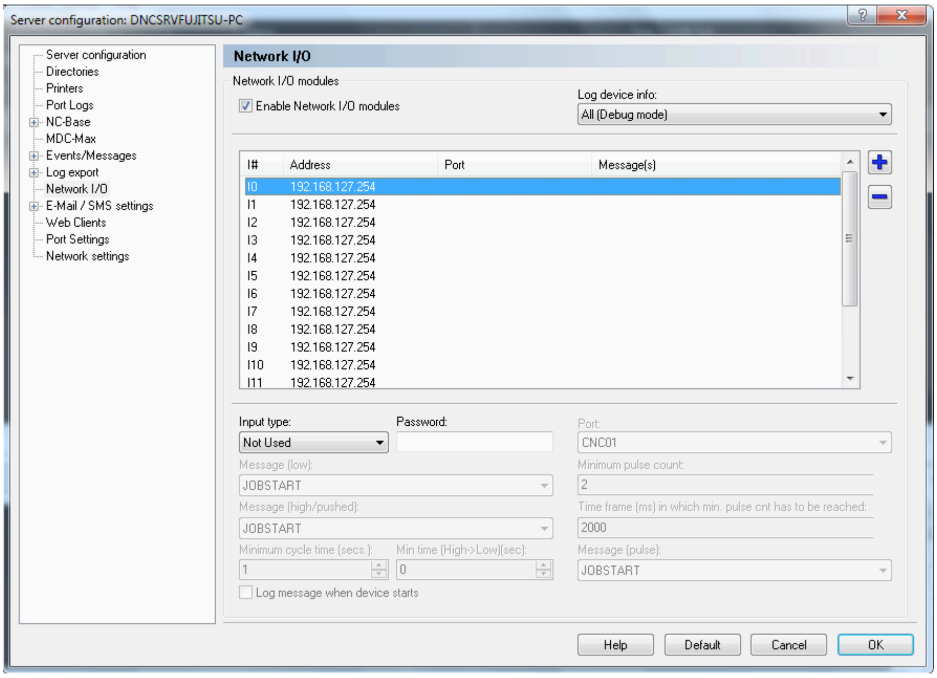
   Contrary to the serial configuration, instead of configuring the messages at each port, the configuration of all network inputs/outputs is gathered in one place. This is more convenient and efficient when managing and configuring many ports/machines of this type.

2. Next, select **Setup / Configure Server** to bring up the DNC-Max Server configuration. Select **Network I/O** from the the treeview menu and check the field **Enable Network I/O modules**. Also, make sure **All (Debug mode)** is selected in the dropdown **Log device info**. That way, all activity on our port is logged.

3. To add the MOXA ioLogik device to the list of network I/O devices, click the [+] icon which brings up the **Add new network I/O device**.

4. In the **Device Address** field enter the IP address of the ioLogik. From the **Device Type** dropdown, select **MOXA E1200**. This is the device series name. Click the **OK** button, and all the inputs on the ioLogik will now show in the list.

   Please note: For this guide you can enter a random IP address, such as 123.456.789.101, if you do not have a MOXA ioLogik E1210 connected.



Server configuration: DNCSRVFUJITSU-PC

- Server configuration
- Directories
- Printers
- Port Logs
- NC-Base
- MDC-Max
- Events/Messages
- Log export
- Network I/O
- E-Mail / SMS settings
- Web Clients
- Port Settings
- Network settings

Network I/O

Network I/O modules

☑ Enable Network I/O modules

Log device info: All (Debug mode)

I#	Address	Port	Message(s)
I0	192.168.127.254		
I1	192.168.127.254		
I2	192.168.127.254		
I3	192.168.127.254		
I4	192.168.127.254		
I5	192.168.127.254		
I6	192.168.127.254		
I7	192.168.127.254		
I8	192.168.127.254		
I9	192.168.127.254		
I10	192.168.127.254		
I11	192.168.127.254		

Input type: Not Used

Password:

Port: CNC01

Message (low): JOBSTART

Minimum pulse count: 2

Message (high/pushed): JOBSTART

Time frame (ms) in which min. pulse cnt has to be reached: 2000

Minimum cycle time (secs.): 1

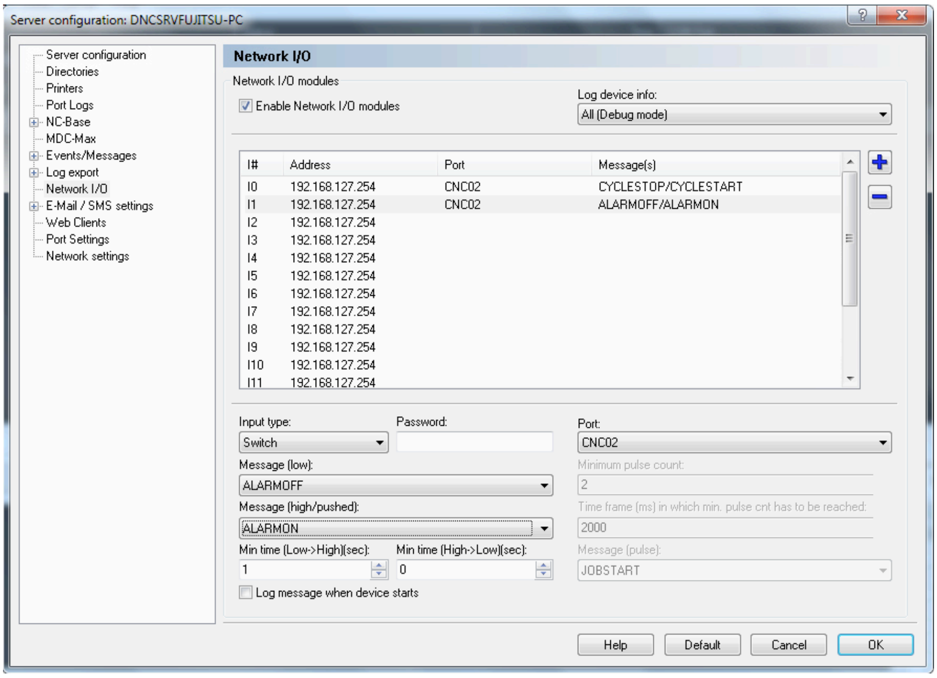
Min time (High->Low)(sec): 0

Message (pulse): JOBSTART

□ Log message when device starts

5. Select the first input **I0**, and set its **Input type** to **Switch**. Set **Message (low)** to CYCLESTOP and **Message (high/pushed)** to CYCLESTART. Finally, set the **Port** to **CNC02**, so the DNC-Max Server knows with which machine this input is associated.

6. Select input **I1**, and set its **Input type** to **Switch**. Set **Message (low)** to ALARMOFF and **Message (high/pushed)** to ALARMON. Finally, set the **Port** to **CNC02**.



Network I/O

Network I/O modules

☑ Enable Network I/O modules

Log device info: All (Debug mode)

I#	Address	Port	Message(s)
I0	192.168.127.254	CNC02	CYCLESTOP/CYCLESTART
I1	192.168.127.254	CNC02	ALARMOFF/ALARMON
I2	192.168.127.254		
I3	192.168.127.254		
I4	192.168.127.254		
I5	192.168.127.254		
I6	192.168.127.254		
I7	192.168.127.254		
I8	192.168.127.254		
I9	192.168.127.254		
I10	192.168.127.254		
I11	192.168.127.254		

Input type: Switch

Password:

Port: CNC02

Message (low): ALARMOFF

Minimum pulse count: 2

Message (high/pushed): ALARMON

Time frame (ms) in which min. pulse cnt has to be reached: 2000

Min time (Low->High)(sec): 1

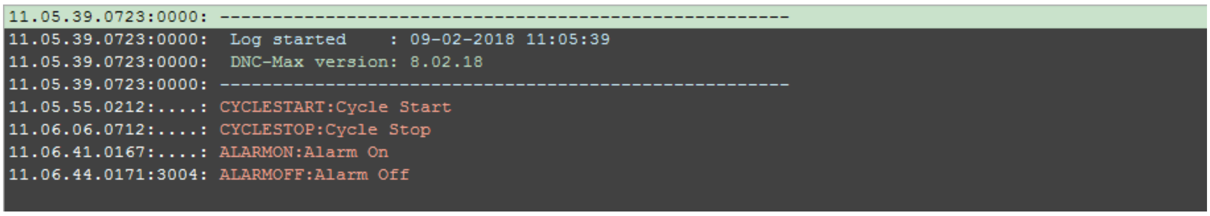
Min time (High->Low)(sec): 0

Message (pulse): JOBSTART

□ Log message when device starts

7. Click the **OK** button to exit the configuration dialog. When asked to restart the server, click **Yes**.

8. From the list of ports, select the new port **CNC02** and click the **Debug tab**. When the machine signals trigger the messages CYCLESTART/STOP and ALARMON/OFF these will show up in the Debug log and we know that the port is configured and the machine is connected properly.



```
11.05.39.0723:0000: -----------------------------------------------------
11.05.39.0723:0000:  Log started     : 09-02-2018 11:05:39
11.05.39.0723:0000:  DNC-Max version: 8.02.18
11.05.39.0723:0000: -----------------------------------------------------
11.05.55.0212:....: CYCLESTART:Cycle Start
11.06.06.0712:....: CYCLESTOP:Cycle Stop
11.06.41.0167:....: ALARMON:Alarm On
11.06.44.0171:3004: ALARMOFF:Alarm Off
```

9. Open or restart the **MDC-Max Client** for the changes to take effect in the MDC-Max Client.

   You will notice that the new machine automatically shows up in the Live Screen and Timeline. Use the **Event Log** to verify that messages from CNC02 are received. Also notice that you now have access to an Operator Screen for both CNC01 and CNC02.

8. Data collection using FANUC FOCAS

Most new CNC machines are equipped with a controller, which supports both serial and network connections. FANUC CNC controllers are some of the most commonly used on CNC machines, and often the data exchange protocol FANUC FOCAS will be available. This protocol provides easy access to all of the CNC's status data and can also be used for many other purposes. FOCAS is short for FANUC Open CNC API Specification.

The third port we will create is one that connects to a machine with a FANUC CNC controller with the FOCAS protocol available. The machine is easily connected to the local network or directly to the test PC using an Ethernet cable, so there is no need for any additional wiring. All we need to know is the IP address of the machine, so we can start configuring the port on the DNC-Max Server.

Please note: For this guide, you do not need a FANUC FOCAS enabled machine to complete the configuration and port messages can still be tested using our Operator Screen.

1. To add the new port, open the **DNC-Max Client**.

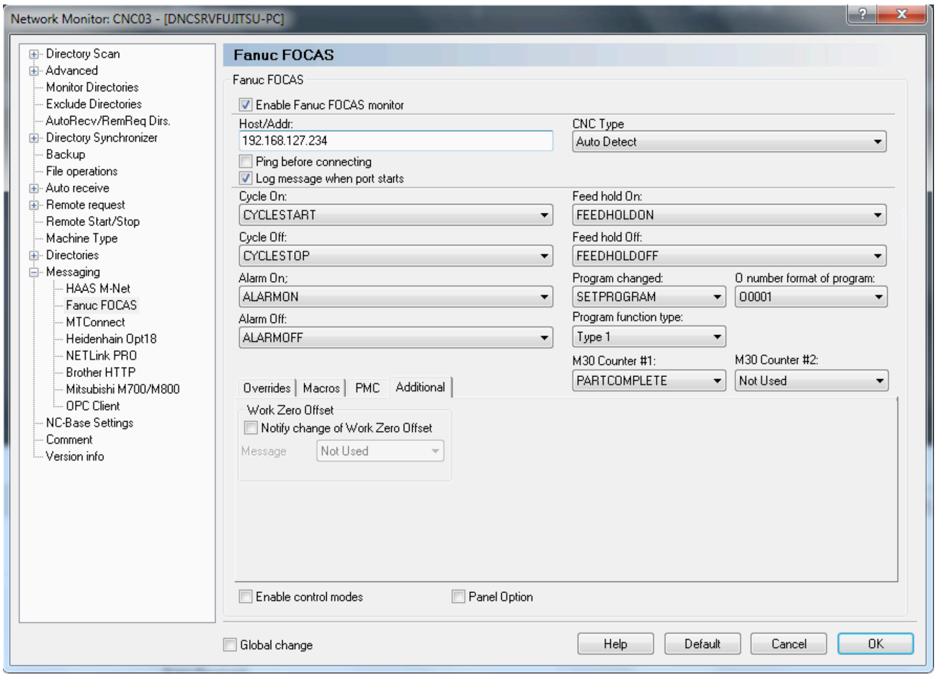
2. Select **Setup** > **Add Port**. In the **Add new machine** dialog, enter the name **CNC03** in the **Description** field and select the Machine template configuration **CNC02**. That way, we use the same template as CNC02 which is the Directory Monitor template. Click **OK** to bring up the **Network Monitor** configuration dialog.

3. In the treeview menu under **Messaging**, select **Fanuc FOCAS**. Check the field **Enable Fanuc FOCAS monitor** to enable the FANUC FOCAS messaging protocol.

4. In the field **Host/Addr** enter the IP address of the machine and make sure **CNC Type** is set to **Auto Detect**. Auto detect will work for all FANUC FOCAS controllers.

   Please note: For this guide, you can enter a random IP address if you do not have an actual machine connected.

   You will notice that the messages for Cycle start/stop and Alarm on/off have already been configured. That is because these are part of the default configured messages for the FANUC FOCAS protocol - and just happens to be the same ones we use in this guide.



Fanuc FOCAS

Fanuc FOCAS

☑ Enable Fanuc FOCAS monitor

Host/Addr: 192.168.127.234

CNC Type: Auto Detect

□ Ping before connecting

☑ Log message when port starts

Cycle On: CYCLESTART

Feed hold On: FEEDHOLDON

Cycle Off: CYCLESTOP

Feed hold Off: FEEDHOLDOFF

Alarm On: ALARMON

Program changed: SETPROGRAM

O number format of program: O0001

Alarm Off: ALARMOFF

Program function type: Type 1

M30 Counter #1: PARTCOMPLETE

M30 Counter #2: Not Used

Work Zero Offset

□ Notify change of Work Zero Offset

Message: Not Used

□ Enable control modes

□ Panel Option

□ Global change

5. Click the **OK** button to close the dialog and notice that the new port shows on in the Port list.

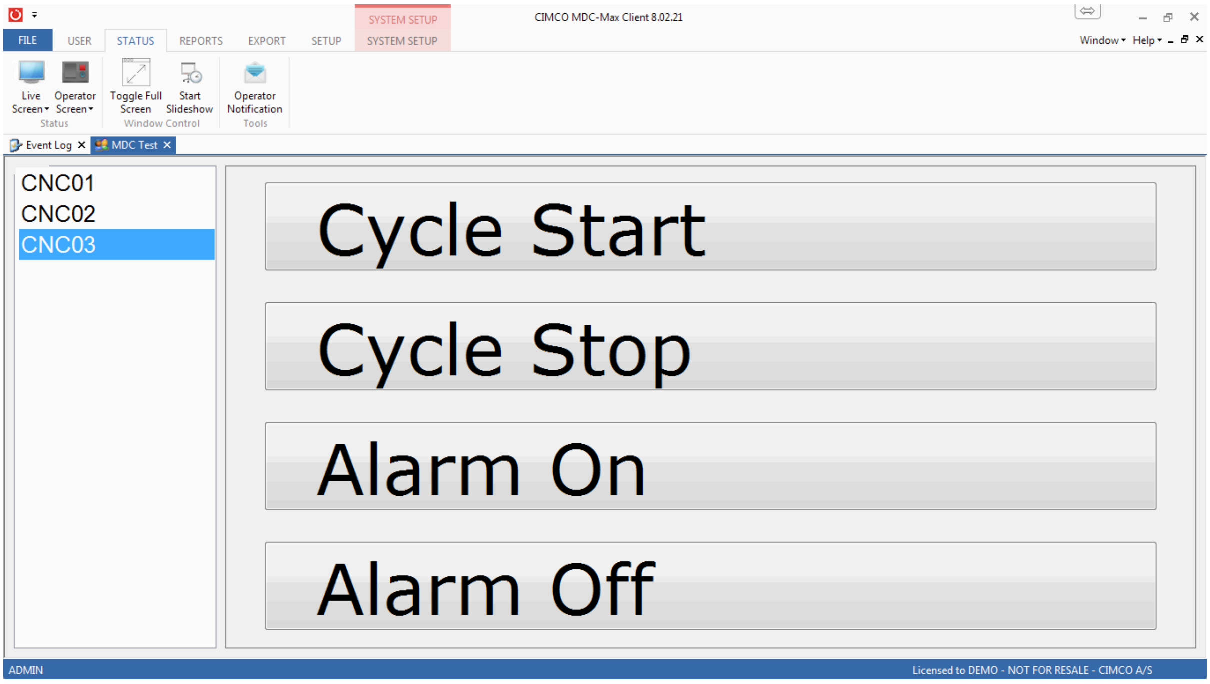
6. Select **Setup** > **Restart server** to make sure the new port as activated properly.

7. Open (or restart) the **MDC-Max Client** to see that the new machine has been added. The machine will automatically show up on the Live Screen and Timeline.

8. To make it easier to switch between the Operator Screens for our different machines we can enable support for multiple machines in the Operator Screen.

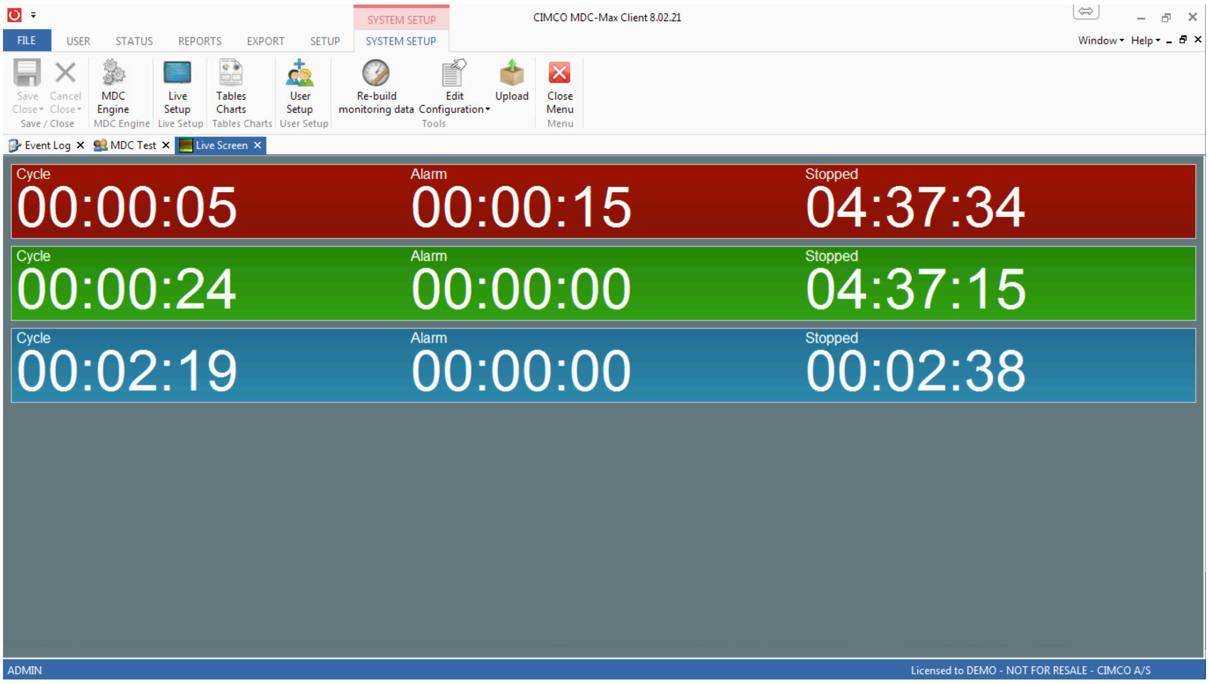
   Select the **Setup tab** and click the **System Setup** button. Click the **Live Setup** button and select **Operator Screen** from the treeview menu. Check the field **Multiple machines on one screen** and click the **Save/Close** button.

9. Reopen the Operator Screen and you will now see a list af the machines on the left side. Select **CNC03** and trigger its messages as needed. Notice how the status of CNC03 changes in the Live Screen and Timeline.



10. The final touch. The text on the Live Screen is too dark and cannot be properly read when viewed from a distance.

    Return to the **Status Screens** configuration, and in the bottom dropdown menu, change the **Text color** to **White**. The Live Screen now looks much better.



9. Associate data with Job and Program number

Until now, our MDC setup has been configured to display the current status of our machines and track the amount of time the machines have been in each state during the day.

In many cases a customer's primary business will be that of supplier or subcontractor which involves running many different jobs during the day. For these customers it is essential that the status and time tracking of each machine get associated with the job they are currently running. Having this type of setup enables the customer to know exactly how long a particular job has taken to complete, but also the amount and type of downtime associated with the job.

MDC-Max has dedicated options for configuring how both job and program numbers should be handled and can associate all the data/messages it receives to the job currently running on each machine automatically.

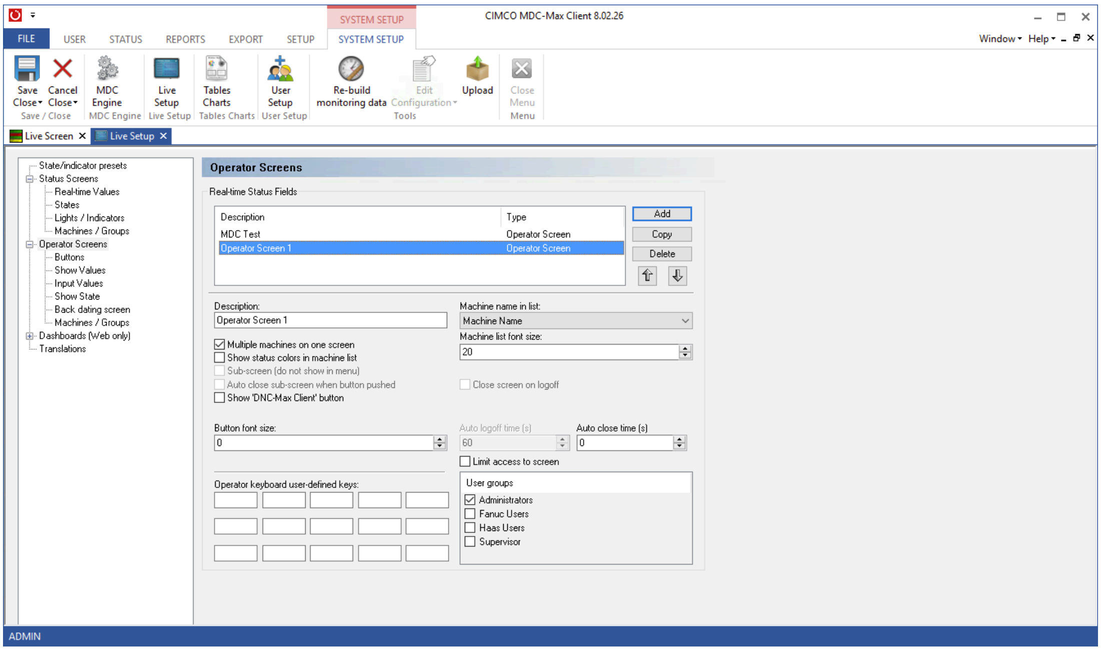
In the following we pretend our customer has a workflow where their machine Operators always receive an Order Sheet containing Job number, Drawing number and Program number. We need to configure MDC-Max and create an Operator Screen that enables Operators to specify the current Job number, Program number and also indicate when a job has been completed.

9.1. Create an Operator Screen with Job, Program and Job End

1. To create a new Operator Screen, bring up the **Live Setup dialog** in the MDC-Max Client by going to **Setup** > **System Setup** > **Live Setup.**

2. In the treeview menu, select **Operator Screens** and click the **Add** button to add a new Operator Screen. In the **Description** field, name the new screen **Operator Screen 1** and enable the option **Multiple machines on one screen**.

   Before continuing to configure the new screen, make sure to select it in the list of Operator Screens.



Operator Screens

Real-time Status Fields

Description	Type
MDC Test	Operator Screen
Operator Screen 1	Operator Screen

Description: Operator Screen 1

Machine name in list: Machine Name

Machine list font size: 20

☑ Multiple machines on one screen
☐ Show status colors in machine list
☐ Sub-screen (do not show in menu)
☐ Auto close sub-screen when button pushed
☐ Show 'DNC-Max Client' button

☐ Close screen on logoff

Button font size: 0

Auto logoff time (s): 60

Auto close time (s): 0

☐ Limit access to screen

Operator keyboard user-defined keys:

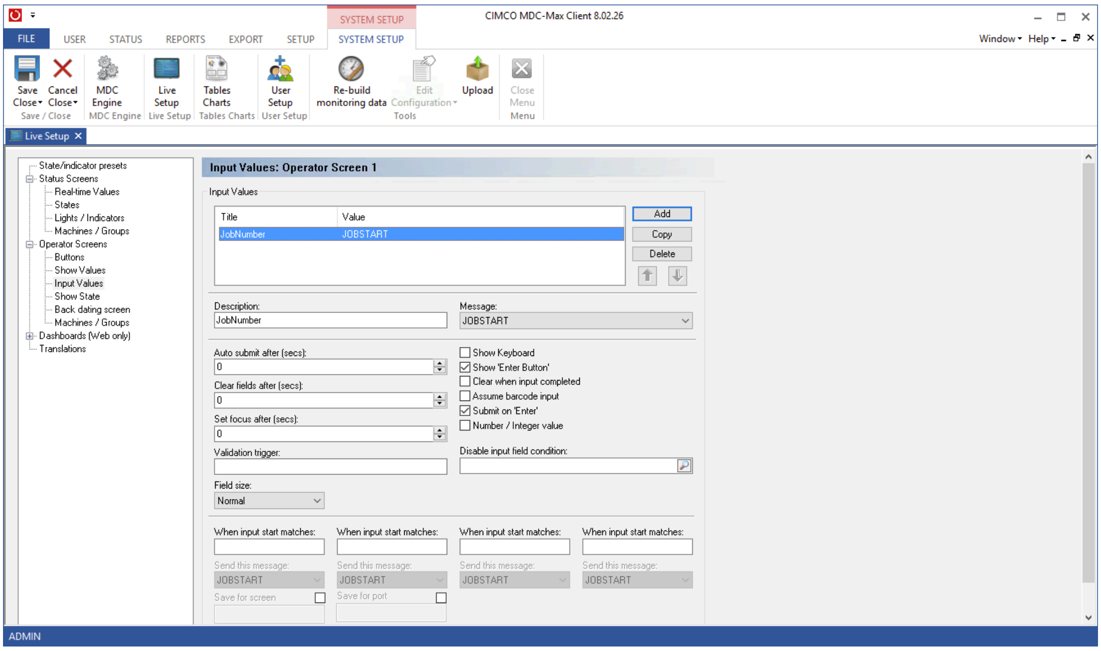
User groups
☑ Administrators
☐ Fanuc Users
☐ Haas Users
☐ Supervisor

3. In the treeview menu, select **Input Values**. This is where input fields can be configured. Notice that the title in the dialog reads **Input Values: Operator Screen 1** indicating that we are configuring input values for Operator Screen 1.

   To add the input field for **JobNumber**, click the **Add** button on the right side of the **Input Values** list. An empty row is inserted in the list and the fields below become active.

   In the **Description** field enter **JobNumber** and from the **Message** dropdown menu select **JOBSTART**. JOBSTART will already be selected because it is the first message in the dropdown menu. So whenever a job number is entered into the field and submitted, a JOBSTART message will be created. The job number will automatically be associated with the message and can be retrieved elsewhere if needed.

   Finally, enable the field **Show 'Enter button'** to add an enter button to the right side of the input field. Also, enable **Submit on 'Enter'** which makes it possible to submit by pressing Enter on the keyboard.



Input Values: Operator Screen 1

Input Values

Title	Value
JobNumber	JOBSTART

Description: JobNumber
Message: JOBSTART

Auto submit after (secs): 0
Clear fields after (secs): 0
Set focus after (secs): 0
Validation trigger:
Field size: Normal

□ Show Keyboard
☑ Show 'Enter Button'
□ Clear when input completed
□ Assume barcode input
☑ Submit on 'Enter'
□ Number / Integer value

Disable input field condition:

When input start matches:
Send this message: JOBSTART
Save for screen □

When input start matches:
Send this message: JOBSTART
Save for port □

When input start matches:
Send this message: JOBSTART

When input start matches:
Send this message: JOBSTART

   Since the input field for ProgramNumber is slightly more complicated, we will first create the button for Job End.

4. To create a button for **Job End** select **Buttons** in the treeview menu. Click the **Add** button on the right side of the **Buttons** list. In the **Description** field enter **Job End** and from the **Push button message** dropdown menu select **JOBSTOP**.

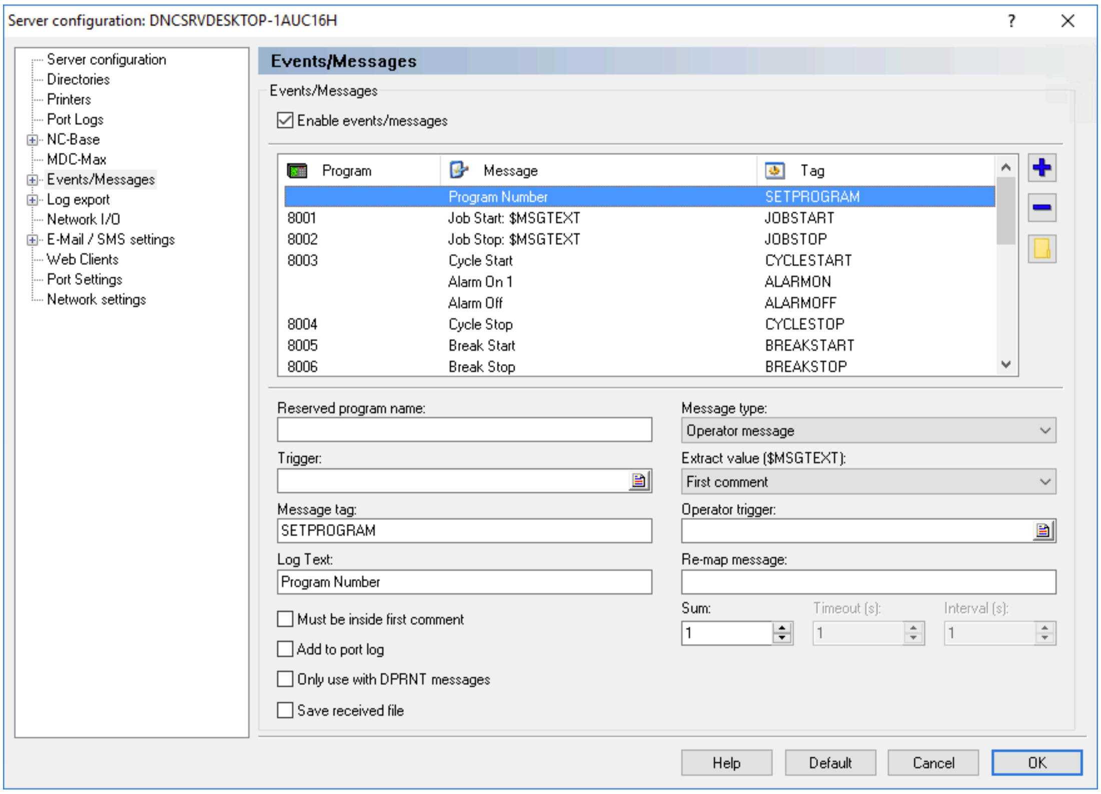
5. Click the **Save / Close** button in the top menu to exit Live Setup.

We now have an input field we can use to enter job number and a Job End button for indicating when a job has been completed. Our setup also requires an input field for ProgramNumber, however, there is no default message we can use for this field. We therefore need to create a new message on the DNC-Max Server called SETPROGRAM.

6. Open the **DNC-Max Client** and select **Setup** > **Configure Server** to bring up the **Server configuration dialog**.

7. In the treeview menu, select **Events/Messages** and click the **[+]** icon to add a new empty message in the list. In the field **Message tag** enter **SETPROGRAM** and in the field **Log**

Text enter **Program Number.** Click the **OK button** to save and exit the configuration.



Server configuration: DNCSRVDESKTOP-1AUC16H

Events/Messages

Events/Messages

☑ Enable events/messages

Program	Message	Tag
	Program Number	SETPROGRAM
8001	Job Start: $MSGTEXT	JOBSTART
8002	Job Stop: $MSGTEXT	JOBSTOP
8003	Cycle Start	CYCLESTART
	Alarm On 1	ALARMON
	Alarm Off	ALARMOFF
8004	Cycle Stop	CYCLESTOP
8005	Break Start	BREAKSTART
8006	Break Stop	BREAKSTOP

Reserved program name:

Message type: Operator message

Trigger:

Extract value ($MSGTEXT): First comment

Message tag: SETPROGRAM

Operator trigger:

Log Text: Program Number

Re-map message:

□ Must be inside first comment

Sum: 1 Timeout (s): 1 Interval (s): 1

□ Add to port log

□ Only use with DPRNT messages

□ Save received file

8. Whenever changes are made to the DNC-Max Server configuration, the server must be restarted for the changes to take effect. Select **Setup** > **Restart Server** and confirm the restart in the dialog that appears.

9. Once the server has restarted, return to the **Live Setup** configuration in the **MDC-Max Client**. If Live Setup is already open in a tab, make sure to reopen it. Otherwise the changes we made on the DNC-Max Server will not show and you will not see the SETPROGRAM message which will obviously be quite confusing.

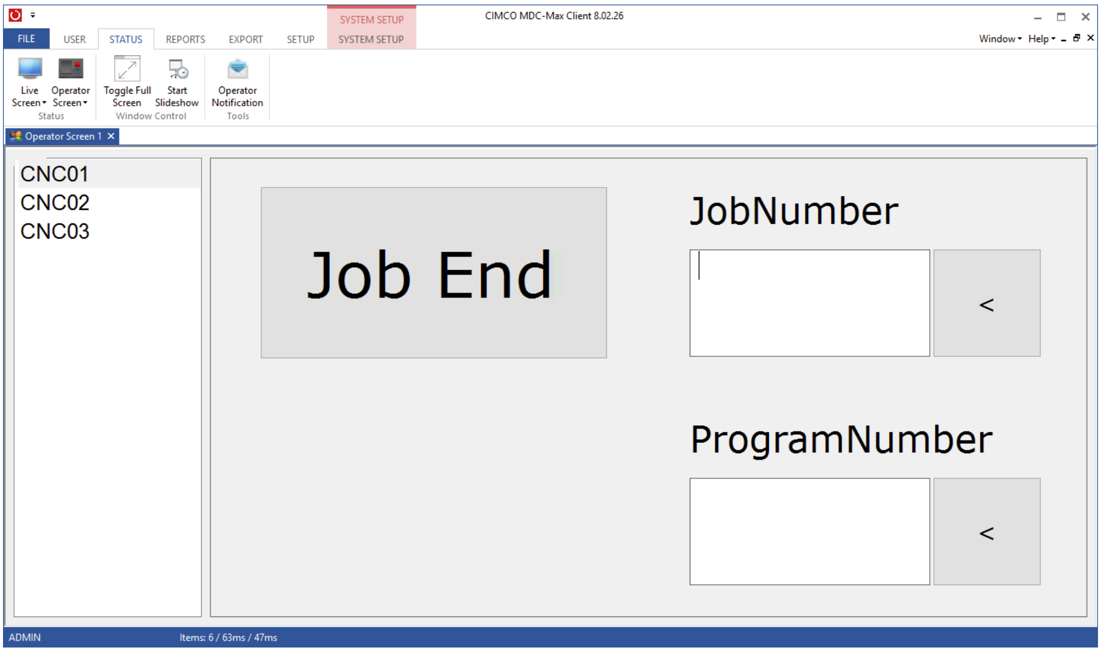
10. From the treeview menu select **Input Values** and make sure the title writes **Input Values: Operator Screen 1**. To add the input field for **ProgramNumber**, click the **Add** button on the right side of the **Input Values** list. The fields below the list then becomes active.

    In the **Description** field enter **ProgramNumber** and from the **Message** dropdown menu select **SETPROGRAM**. Enable the fields **Show 'Enter button'** and **Submit on 'Enter'**.

11. Click the **Save / Close** button in the top menu to exit Live Setup.

12. To test the new Operator Screen, click the button **Status** > **Operator Screen**. If you click the button-text and not the icon, you can choose which Operator Screen to show.

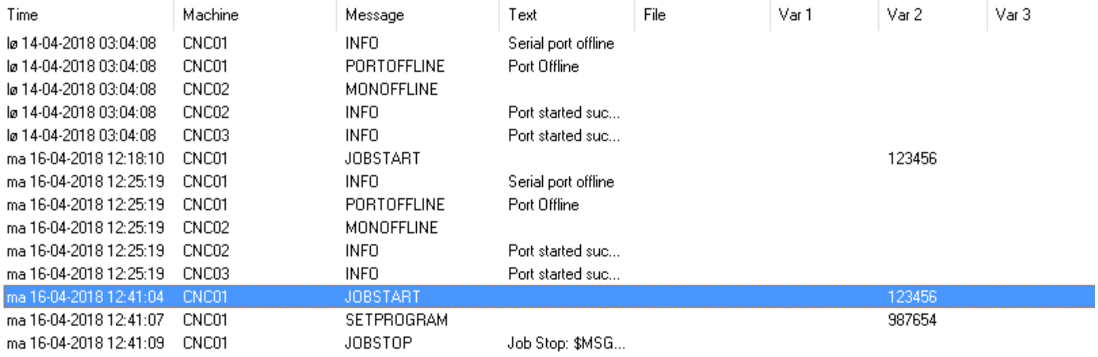
    As you can see the screen contains input fields for JobNumber, ProgramNumber and the Job End button.



Enter a random job and program number and make sure to submit each field by clicking the submit button or by pressing Enter on the keyboard while a field is active. Also, click the Job End button.

13. To verify that the messages associated with the input fields and button are working, select **Reports** > **Event Log**. In the Event Log, you should see log entries for the messages **JOBSTART, SETPROGRAM** and **JOBSTOP**.

    Notice that JOBSTART and SETPROGRAM both have values associated in the column **Var 2**. These are the job number and program number you just entered using the Operator Screen.



Time	Machine	Message	Text	File	Var 1	Var 2	Var 3
lø 14-04-2018 03:04:08	CNC01	INFO	Serial port offline				
lø 14-04-2018 03:04:08	CNC01	PORTOFFLINE	Port Offline				
lø 14-04-2018 03:04:08	CNC02	MONOFFLINE					
lø 14-04-2018 03:04:08	CNC02	INFO	Port started suc...				
lø 14-04-2018 03:04:08	CNC03	INFO	Port started suc...				
ma 16-04-2018 12:18:10	CNC01	JOBSTART				123456	
ma 16-04-2018 12:25:19	CNC01	INFO	Serial port offline				
ma 16-04-2018 12:25:19	CNC01	PORTOFFLINE	Port Offline				
ma 16-04-2018 12:25:19	CNC02	MONOFFLINE					
ma 16-04-2018 12:25:19	CNC02	INFO	Port started suc...				
ma 16-04-2018 12:25:19	CNC03	INFO	Port started suc...				
ma 16-04-2018 12:41:04	CNC01	JOBSTART				123456	
ma 16-04-2018 12:41:07	CNC01	SETPROGRAM				987654	
ma 16-04-2018 12:41:09	CNC01	JOBSTOP	Job Stop: $MSG...				

9.2. Configure Job and Program in the MDC Engine

With our Operator Screen in place and messages being successfully logged we now need to configure MDC-Max so it "knows" from where to retrieve the job and program numbers. Since handling of job and program is a core part of MDC-Max, this is configured in the MDC Engine.

1. In the **MDC-Max Client**, bring up the **MDC Engine** configuration by going to **System** > **System Setup** > **MDC Engine**.

2. First we will configure the Program number. Select **Program Setup** in the treeview menu.

   The topmost list field (right under the title **Extracting Program Name**) is where we can add the message from where the program name (or number in our case) should be extracted.

   If the row MSG[PROGRAMSTART] is already in the list, delete it or modify it as described next.

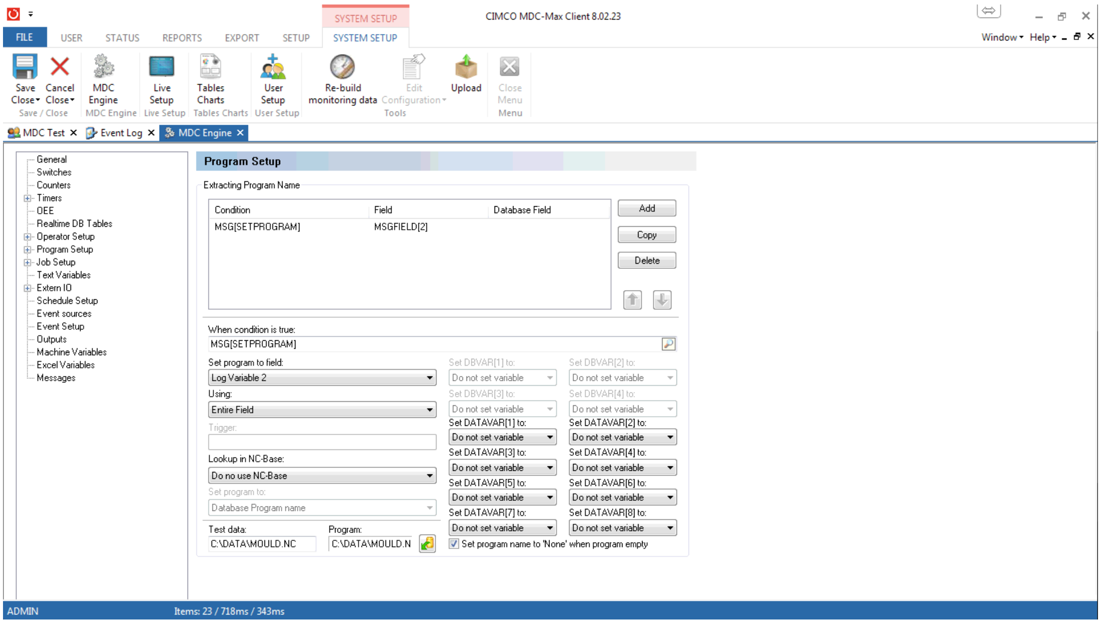
3. Click the **Add** button on the right side of the list. The fields below the list then becomes active. In the field **When condition is true** click the small search icon or double-click to bring up the **Expression dialog.** Locate and select **MSG[SETPROGRAM]** and click the **Insert** button so it appears in the **Expression box**. Then click the **OK** button.

   The field **When condition is true** determines under what condition MDC-Max should extract and set a new program name. In our case the condition becomes true whenever a SETPROGRAM message is generated, however, more complicated conditions can be defined if required.

4. In the field **Set program to field**, select **Log Variable 2**. That way, MDC-Max retrieves the program number from the column **Var 2** which we saw earlier in the Event log contains the program number associated with the SETPROGRAM message.

   Note that the field **Using** specifies how the value in Log Variable 2 (the **Var 2** column) should be extracted. In our case we leave it set to **Entire field**, which grabs the entire value in the field.

5. At the bottom of the dialog, enable **Set program name to 'None' when program empty**. This ensures that if no program name (or number in our case) has been specified, the program name will be set to the value **None** instead of nothing/blank. This makes it easier to see if something is wrong when displaying the program name on a Live Screen or elsewhere.



Program Setup

Extracting Program Name

Condition	Field	Database Field
MSG[SETPROGRAM]	MSGFIELD[2]	

When condition is true: MSG[SETPROGRAM]

Set program to field: Log Variable 2

Using: Entire Field

Trigger:

Lookup in NC-Base: Do no use NC-Base

Set program to: Database Program name

Test data: C:\DATA\MOULD.NC

Program: C:\DATA\MOULD.N

Set DBVAR[1] to: Do not set variable

Set DBVAR[2] to: Do not set variable

Set DBVAR[3] to: Do not set variable

Set DBVAR[4] to: Do not set variable

Set DATAVAR[1] to: Do not set variable

Set DATAVAR[2] to: Do not set variable

Set DATAVAR[3] to: Do not set variable

Set DATAVAR[4] to: Do not set variable

Set DATAVAR[5] to: Do not set variable

Set DATAVAR[6] to: Do not set variable

Set DATAVAR[7] to: Do not set variable

Set DATAVAR[8] to: Do not set variable

☑ Set program name to 'None' when program empty

6. Next, we need to configure the job number in a similar manner. Select **Job Setup** in the treeview menu.

If the row MSG[JOBSTART] is already in the list, delete it or modify it as described next.

Click the **Add** button and in the field **When condition is true** locate and insert **MSG[JOBSTART]**. In the field **Set program to field**, select **Log Variable 2** and finally enable **Set program name to 'None' when program empty**.

7. Click the **Save / Close** button in the top menu to exit the MDC Engine configuration. MDC-Max now knows from where to retrieve the job and program numbers.

9.3. Display Job and Program number on Live Screen and Timeline

With the job and program numbers all set up we will now include these on our Live Screen and Timeline.

1. To configure our existing Live Screen, bring up the **Live Setup** dialog in the MDC-Max Client by going to **Setup** > **System Setup** > **Live Setup.**

2. In the treeview menu under **Status Screens**, select **Real-time Values**. This is where we configure the values we want to display on the Live Screen. Make sure you are configuring the Live Screen we created earlier and that this is selected in Status Screens.

3. First we will add the job number. Click the **Add** button and then click on the **search icon** in the **Value** field. This brings up the **Expression dialog**. Find and double-click **TEXT[JOB]**. Click **OK** when done.

   JOB is a predefined variable in MDC-Max which contains the current job name/number associated with a machine. Because MDC-Max knows where to get the job number, this variable can now be used and will automatically contain the current job number. The **TEXT[ ]** part of the expression tells MDC-Max to get the value from the JOB variable as plain text. A similar variable is available for PROGRAM.

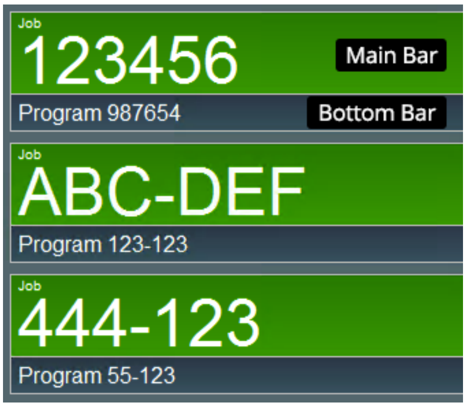
   Finally, in the field **Title** enter **Job** and make sure **Format** is set to **TEXT**.

4. Next, add the program number. Click the **Add** button and then click on the **search icon** in the **Value** field. This brings up the **Expression dialog**. Find and double-click **TEXT[PROGRAM]**. Click **OK** when done.

   In the field **Title** enter **Program** and make sure **Format** is set to **TEXT**.

5. We want the job and program numbers to be placed as the first information from the left. At the top of the dialog in the list **Real-time Status Fields** select **Job** and use the arrow buttons on the right to move it to the top. Next, select **Program** in the list and move it up below **Job**.

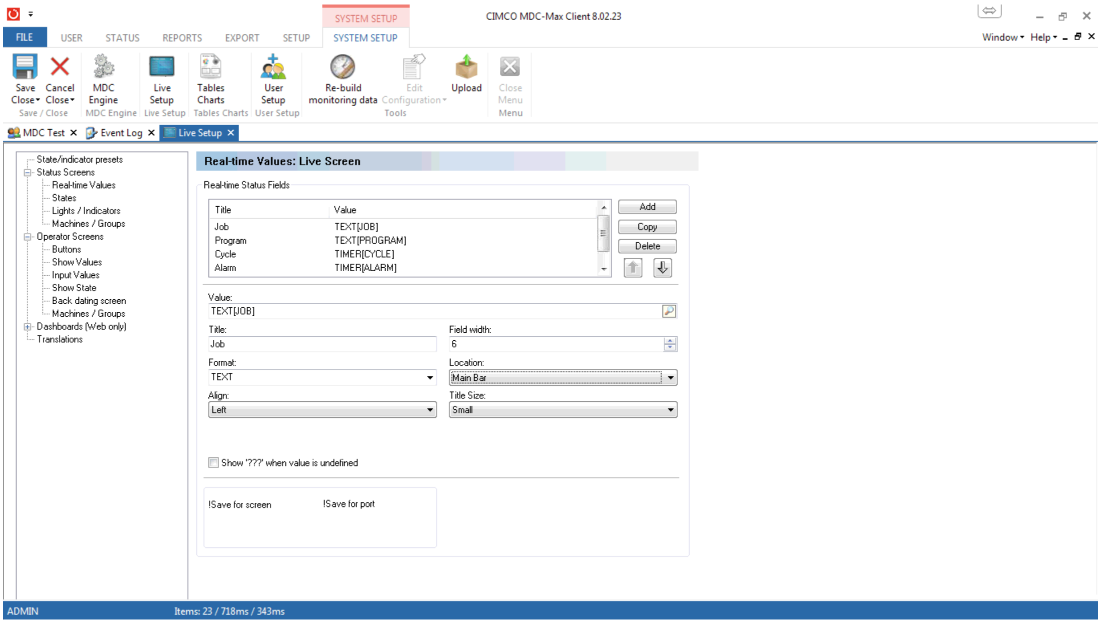
6. Since the program number is of less importance on the Live Screen we will place it below the job number in the **Bottom Bar**. For **Program**, set the **Location** dropdown menu to **Bottom Bar**.



So when the Live Screen is generated MDC-Max will go through the list from the top and will place the first value (Job) first in the Main Bar. The second value (Program) will be placed first in Bottom Bar, the third value Cycle will be placed second in the Main bar and so forth.

MDC-Max prefers to have the Main Bar values ordered first in the list and then the Bottom Bar values. MDC-Max will automatically sort the values when the configuration is saved.

Also, notice that you can use the field **Field width** to adjust the width of each field as needed.



Real-time Values: Live Screen

Real-time Status Fields

Title	Value
Job	TEXT[JOB]
Program	TEXT[PROGRAM]
Cycle	TIMER[CYCLE]
Alarm	TIMER[ALARM]

Value: TEXT[JOB]

Title: Job

Field width: 6

Format: TEXT

Location: Main Bar

Align: Left

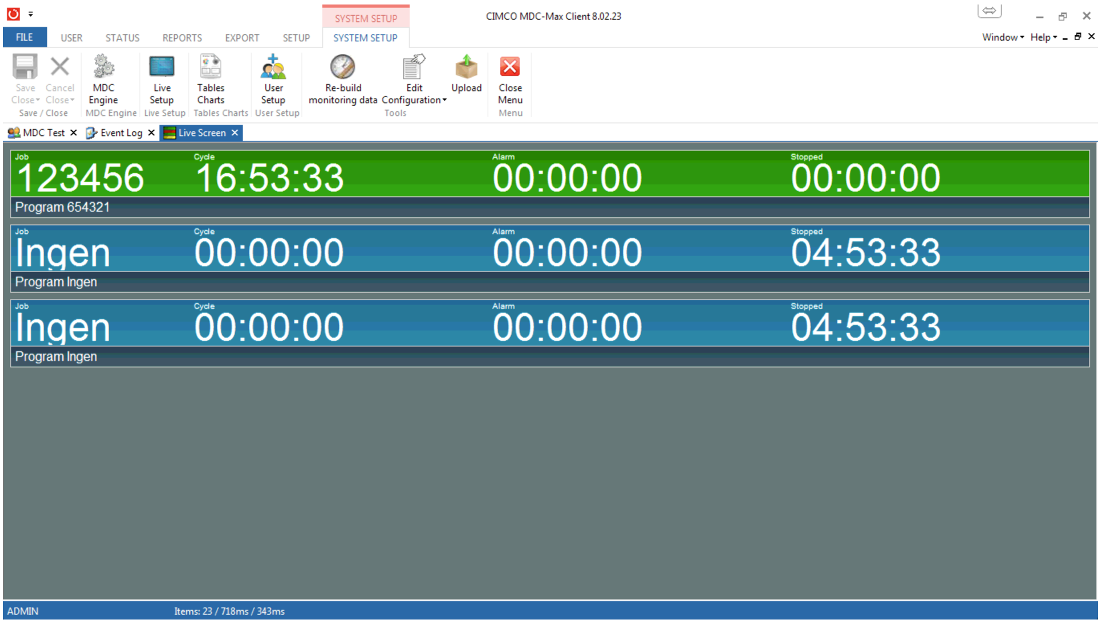
Title Size: Small

□ Show '???' when value is undefined

!Save for screen

!Save for port

7. When done, click the **Save/Close** button in the top menu. Bring up the **Live Screen** by going to **Status** > **Live Screen**. Use the Operator Screens we created to change machine state and enter job and program numbers to see how the data is reflected in the Live Screen.

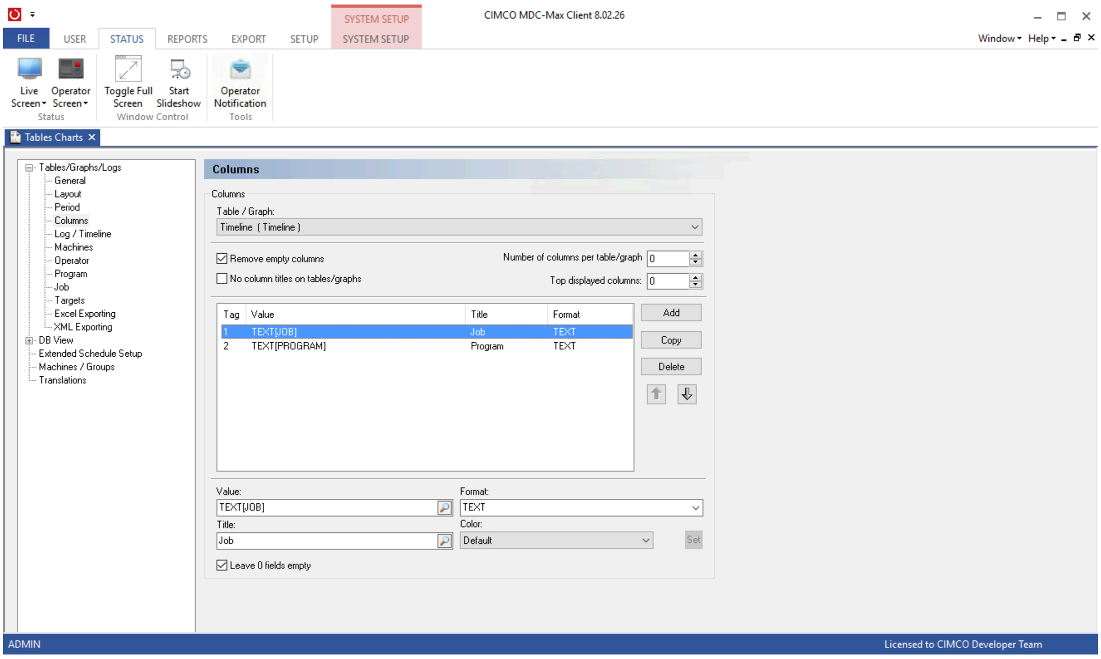


Job	Cycle	Alarm	Stopped
123456	16:53:33	00:00:00	00:00:00
Program 654321			
Ingen	00:00:00	00:00:00	04:53:33
Program Ingen			
Ingen	00:00:00	00:00:00	04:53:33
Program Ingen			

8. We also want to show the job and program numbers on our Timeline. To do this, open the **Tables/Charts configuration** by going to **Setup** > **System Setup** > **Tables/Charts.**

9. In the treeview menu, select **Columns**. Click the **Add** button and add the expression **TEXT[JOB]**. In the **Title** field enter **Job** and in the **Format** dropdown menu select **TEXT**.

10. Next we will add the program number. Click the **Add** button and add the expression **TEXT[PROGRAM]**. In the **Title** field enter **Program** and in the **Format** dropdown menu select **TEXT**.



Columns

Columns

Table / Graph: Timeline (Timeline)

☑ Remove empty columns

Number of columns per table/graph: 0

☐ No column titles on tables/graphs

Top displayed columns: 0

Tag	Value	Title	Format
1	TEXT[JOB]	Job	TEXT
2	TEXT[PROGRAM]	Program	TEXT

Value: TEXT[JOB]

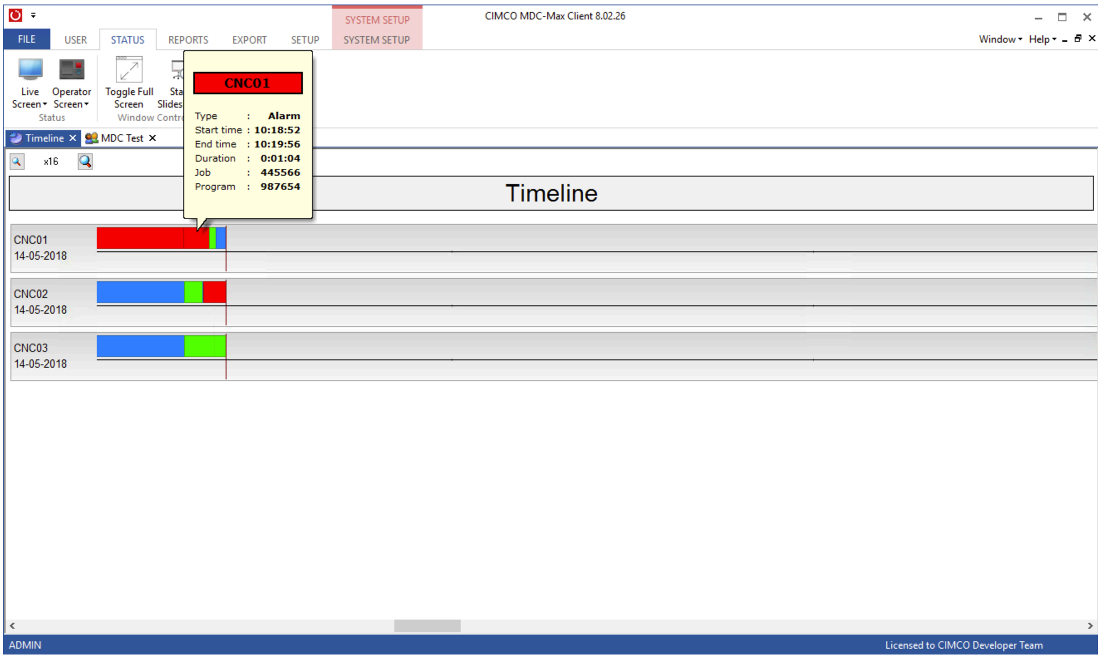
Format: TEXT

Title: Job

Color: Default

☑ Leave 0 fields empty

11. Click the **Save/Close** button in the top menu and open the Timeline by going to **Reports** > **Generate Graph**. Click the **Today** button to show the Timeline from today and click the **OK** button. The job and program number is shown in the tooltip that appears when holding the mouse cursor over a machine timeline.



CNC01

Type : Alarm
Start time : 10:18:52
End time : 10:19:56
Duration : 0:01:04
Job : 445566
Program : 987654

9.4. Create and display Job Timer on Live Screen

In order to measure the time from job start to completion we need to create a dedicated Job Timer which we will also display on the Live Screen. Our existing Timers (CYCLE, STOPPED and ALARM) also have to be modified so they are reset when a new job is started instead of at the end of the day. That way all the information on the Live Screen will be related to the current job.

1. In the **MDC-Max Client**, open the **MDC Engine** configuration by going to **System** > **System Setup** > **MDC Engine**.

2. In the treeview menu select **Timers** and click the **Add** button to create a new Timer. In the **New Timer Name** dialog that appears, name the Timer **JOBTIMER** and click **OK** to add it to the list.

3. Select the **JOBTIMER** in the list and below in the field **Description** enter **Timer JOBTIMER**.

4. Set **Start condition** to **MSG[JOBSTART]** and **Stop condition** to **MSG[JOBSTOP]** since we want the Timer to start when the JOBSTART message is received and stop when the JOBSTOP message is received. Finally, set the **Clear condition** to **MSG[JOBSTART]** so the timer is reset when a new job is started. In this way our Live Screen will retain the job related information when a job has been completed and only when a new job is started will the Timers be reset.

5. For the other Timers in the list (CYCLE, STOPPED and ALARM), change their **Clear condition** to **MSG[JOBSTART]**. In this way these timers are reset when a new job is started.

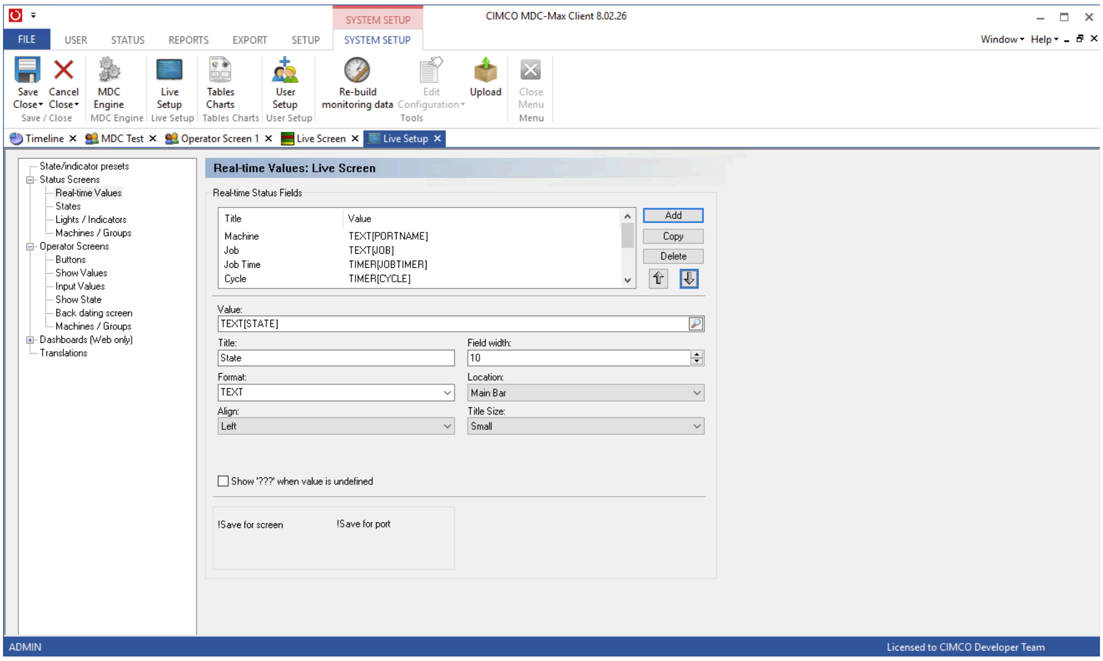
6. Click the **Save / Close** button in the top menu to exit the MDC Engine configuration.

7. Next we wil add the Job Timer to the Live Screen. Bring up the **Live Setup dialog** in the MDC-Max Client by going to **Setup** > **System Setup** > **Live Setup.** In the treeview menu under **Status Screens**, select **Real-time Values**.

8. Click the **Add** button and then click on the **search icon** in the **Value** field. This brings up the **Expression dialog**. Find and double-click **TIMER[JOBTIMER]**. Click **OK** when done.

   In the field **Title** enter **Job Time** and set the **Format** to **HHMS**.

   Since we want the timer to be displayed after the job number set **Location** to **Main Bar** and use the arrows to move **Job Time** up below **Job** in the list.



Real-time Values: Live Screen

Real-time Status Fields

Title	Value
Machine	TEXT[PORTNAME]
Job	TEXT[JOB]
Job Time	TIMER[JOBTIMER]
Cycle	TIMER[CYCLE]

Value: TEXT[STATE]

Title: State

Field width: 10

Format: TEXT

Location: Main Bar

Align: Left

Title Size: Small

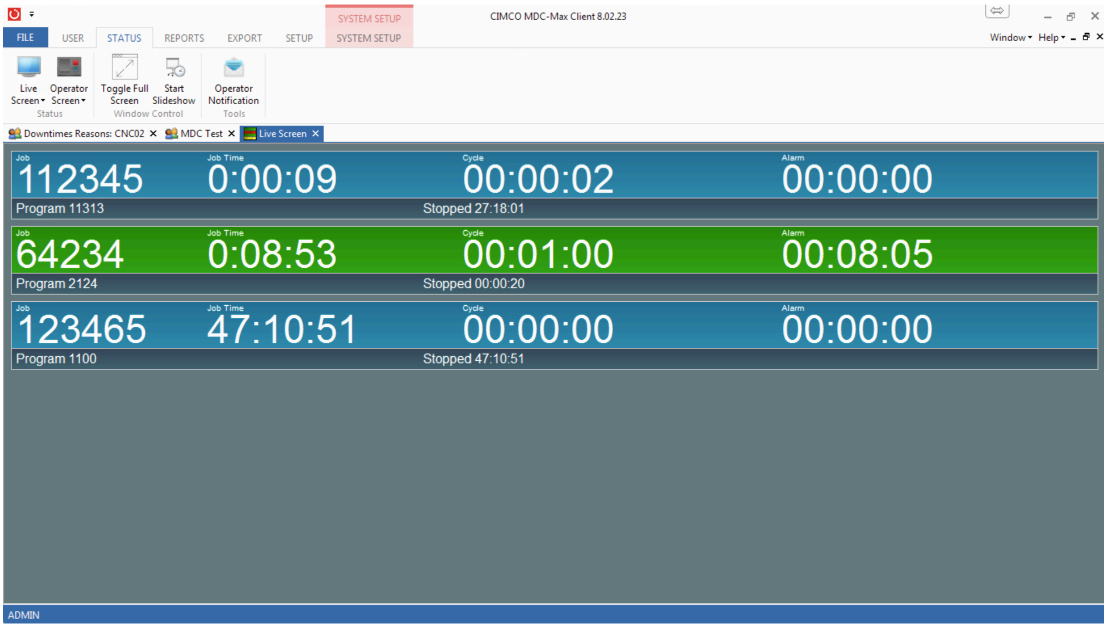
□ Show '???' when value is undefined

!Save for screen

!Save for port

9. To make more room in the Main Bar we will display the Stopped time in Bottom Bar 1 instead of in the Main Bar. Select **Stopped** in the list and set its **Location** to **Bottom Bar 1**.

10. Click the **Save/Close** button in the top menu. Bring up the **Live Screen** by going to **Status** > **Live Screen**. Use the Operator Screens we created to change machine state and enter job and program numbers, and see how the data is reflected in the Live Screen.



Job	Job Time	Cycle	Alarm
112345	0:00:09	00:00:02	00:00:00
Program 11313		Stopped 27:18:01	
64234	0:08:53	00:01:00	00:08:05
Program 2124		Stopped 00:00:20	
123465	47:10:51	00:00:00	00:00:00
Program 1100		Stopped 47:10:51	

10. Downtime reasons and state

Detailed knowledge about why machines are not running is essential to understanding machine downtime, utilization problems, and to increase shop floor productivity. As we have seen, MDC-Max can automatically collect the accurate and error-free machine data, however to fully understand the reasons for downtime, the contextual information that only Operators can provide is needed.

By predefining the possible reasons for downtime and enabling Operators to easily indicate these to MDC-Max, it is possible to see exactly how a machine and Operator has performed during a specific job or time period.

In the following we will expand our Operator Screen 1 to include buttons for indicating different downtime reasons. Each button will trigger a dedicated Timer that measures for how long a machine has been down for that specific reason. We will also update our Operator Screen, Live Screen and Timeline to include information about machine state and downtime reason.

10.1. Create Messages, Timers and Timer Group

In the following we will create the messages that will be triggered by the downtime buttons and the Timers that will be activated by those messages.

1. First the messages. Open the **DNC-Max Client** and from the **Setup** menu select **Configure Server** to bring up the **Server configuration dialog**. In the treeview menu, select **Events/Messages.**

2. Add the five messages below:

Message tag	**Log text**
SETTING	Setting
TOOLING	Tooling
COOLINGFLUID	Cooling Fluid
CHIPS	Chips
OKTORUN	OK To Run

   The OKTORUN message will be used for indicating when an issue, that causes downtime, has been resolved and the machine is *OK to run*.

3. When done, click the **OK button** to exit the configuration.

4. From the **Setup** menu select **Restart Server** and confirm the restart in the dialog that appears.

5. Next we will create the Timers. Open the **MDC-Max Client** and open the **MDC Engine** configuration by going to **System** > **System Setup** > **MDC Engine**. In the treeview menu select **Timers**.

6. Add the five Timers below.

   Notice that we do not set a Stop condition for these Timers. We will set a common Stop condition using a Timer Group afterwards. You will likely get a dialog telling you that "No stop and/or reset condition specified!" - just ignore that dialog whenever it shows.

Timer name	Start condition	Clear condition
SETTING	MSG[SETTING]	MSG[JOBSTART]
TOOLING	MSG[TOOLING]	MSG[JOBSTART]
COOLINGFLUID	MSG[COOLINGFLUID]	MSG[JOBSTART]
CHIPS	MSG[CHIPS]	MSG[JOBSTART]
OKTORUN	MSG[OKTORUN]	MSG[JOBSTART]

7. In the treeview menu below **Timers**, select **Timer Groups**. If you do not see **Timer Groups**, expand the tree by clicking the small [+] icon on the left side of Timers.

   Timer Groups lets you group related Timers and specify common properties such as a common Stop condition. Timers in a group can be mutually exclusive, meaning that only one of the Timers can be running at a time. Also, Timer Groups can be included in reports to show e.g. the amount of downtime for a machine or you might group Timers by planned and unplanned downtime. In the following we will create a single group for our downtime Timers.

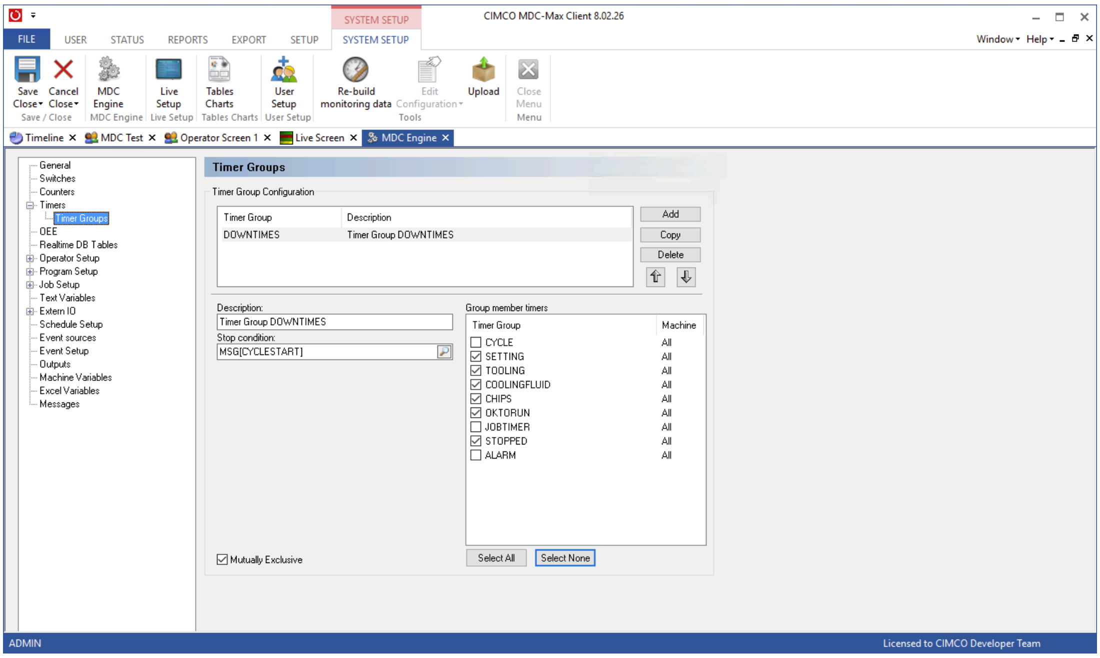
8. To create a new Timer Group, click the **Add** button on the right side of the list. In the dialog that appears enter the name **DOWNTIMES**.

9. Below, in the list **Group member timers** select the following timers. These are all the Timers that can be active when a machine has stopped:

   SETTING
   COOLINGFLUID
   CHIPS
   TOOLING
   STOPPED
   OKTORUN

10. In the field **Stop condition** set **MSG[CYCLESTART]** and at the bottom of the dialog enable the field **Mutually Exclusive**.

    So if any one of the Timers in our DOWNTIMES group is running it will stop when a CYCLESTART message is received which of course signifies that the machine is running again.


CIMCO MDC-Max Client 8.02.26
FILE USER STATUS REPORTS EXPORT SETUP SYSTEM SETUP
Save Close Cancel Close MDC Engine Live Setup Tables Charts User Setup Re-build monitoring data Edit Configuration Upload Close Menu
Timeline MDC Test Operator Screen 1 Live Screen MDC Engine
General
Switches
Counters
Timers
Timer Groups
OEE
Realtime DB Tables
Operator Setup
Program Setup
Job Setup
Text Variables
Extern IO
Schedule Setup
Event sources
Event Setup
Outputs
Machine Variables
Excel Variables
Messages
ADMIN
Licensed to CIMCO Developer Team

Timer Groups

Timer Group Configuration

Timer Group	Description
DOWNTIMES	Timer Group DOWNTIMES

Add · Copy · Delete

Description: Timer Group DOWNTIMES

Stop condition: MSG[CYCLESTART]

☑ Mutually Exclusive

Group member timers

Timer Group	Machine
□ CYCLE	All
☑ SETTING	All
☑ TOOLING	All
☑ COOLINGFLUID	All
☑ CHIPS	All
☑ OKTORUN	All
□ JOBTIMER	All
☑ STOPPED	All
□ ALARM	All

Select All · Select None

11. To see what group settings have been configured for a specific Timer, go to **Timers** in the treeview and select for instance the Timer CHIPS. At the bottom of the dialog, click the button **View Group Effects**. In the dialog that appears you can see how the Timer's Stop condition is configured. This can be very handy when troubleshooting.

    It will look like this:

    (MSG[SETTING]) OR (MSG[TOOLING]) OR (MSG[COOLINGFLUID]) OR (MSG[CYCLESTOP]) OR (MSG[CYCLESTART]).

    As you can see, any of these messages will cause the CHIPS Timer to stop as defined in our Timer Group.

12. Click the **Save/Close** button in the top menu.

10.2. Add downtime buttons and state to Operator Screen

In the following we will add five downtime buttons and a state indicator to Operator Screen 1.

1. Bring up the **Live Setup dialog** in the MDC-Max Client by going to **Setup** > **System Setup** > **Live Setup.** In the treeview menu, select **Operator Screens** and in the dialog make sure **Operator Screen 1** is selected in the list.

2. Next, select **Buttons** in the treeview menu and add the following five buttons:

Description	Push button message	Enable Show indicator	Color	Value
Setting	SETTING	Enable	Red	TIMERRUNS[SETTING]
Tooling	TOOLING	Enable	Red	TIMERRUNS[TOOLING]
Cooling Fluid	COOLINGFLUID	Enable	Red	TIMERRUNS[COOLINGFLUID]
Chips	CHIPS	Enable	Red	TIMERRUNS[CHIPS]
Ok To Run	OKTORUN	Enable	Green	TIMERRUNS[OKTORUN]

3. Next, we will add a state indicator by selecting **Show State** In the treeview menu. State on an Operator Screen is always shown at the top in an area that spans the entire screen.

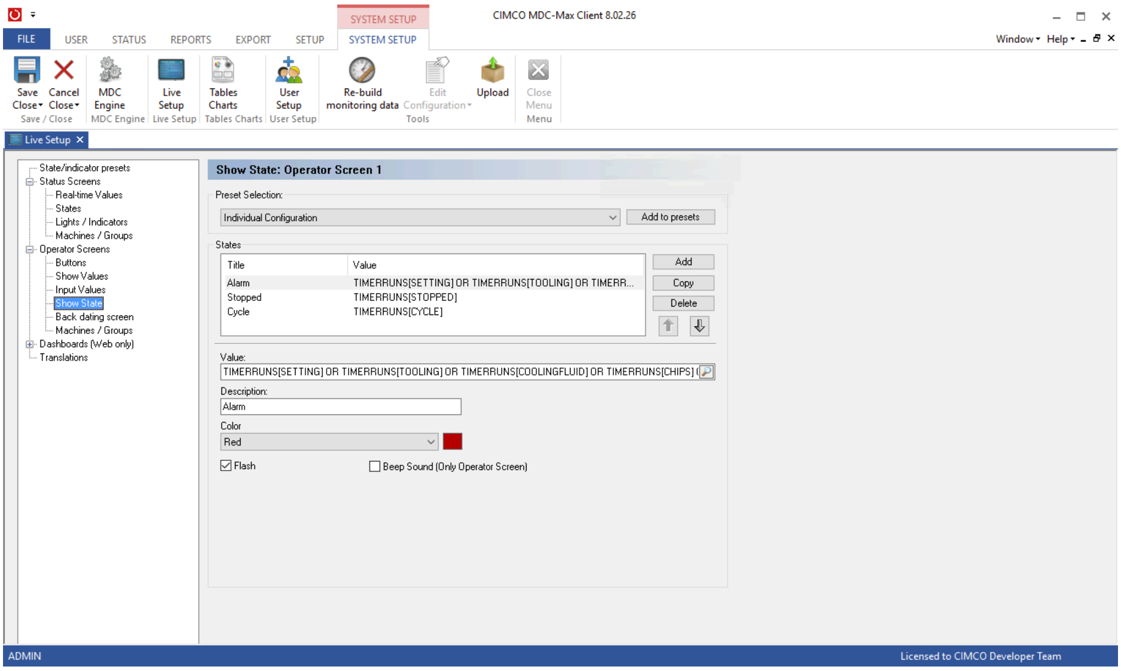
   A state is triggered whenever its expression becomes true. You will notice that the **States** list is currently empty, so we need to add the states we want to show.

4. Click the **Add** button on the right side of the list. This will add an empty row and the fields below the list will become active. Add the following three states in the same prioritized order.

   Notice that the Alarm state is defined by a more complex expression. This simply defines that if any of the downtime timers or the Alarm timer is running this will cause the Alarm state expression to become true and thereby trigger the Alarm state.

Value (expression)	**Description**	**Color**	**Flash**	**Beep Sound**
TIMERRUNS[SETTING] OR TIMERRUNS[TOOLING] OR TIMERRUNS[COOLINGFLUID] OR TIMERRUNS[CHIPS] OR TIMERRUNS[ALARM]	Alarm	Red	Enable	-
TIMERRUNS[STOPPED]	Stopped	Grey	-	-
TIMERRUNS[CYCLE]	Cycle	Green	-	-

5.



Show State: Operator Screen 1

Preset Selection: Individual Configuration

States

Title	Value
Alarm	TIMERRUNS[SETTING] OR TIMERRUNS[TOOLING] OR TIMERR...
Stopped	TIMERRUNS[STOPPED]
Cycle	TIMERRUNS[CYCLE]

Value: TIMERRUNS[SETTING] OR TIMERRUNS[TOOLING] OR TIMERRUNS[COOLINGFLUID] OR TIMERRUNS[CHIPS] (

Description: Alarm

Color: Red 🟥

☑ Flash

□ Beep Sound (Only Operator Screen)

   Click the **Save/Close** button in the top menu.

6. Now, open the Operator Screens **Operator Screen 1** and **MDC Test**.

   To simulate an alarm on the machine CNC01, open **MDC Test**, select **CNC01** and click **Cycle Stop** and then **Alarm On**.

   Next, open **Operator Screen 1** and see how the state has changed to Alarm. Use **MDC Test** to simulate other machine states and see how state is reflected in **Operator Screen 1**.

10.3. Show state on Live Screen and Timeline

In the following we will modify our Live Screen and Timeline to also include machine state.

1. In the MDC-Max Client, go to **System** > **System Setup** > **Live Setup** and in the treeview menu select **Real-time Values**.

2. On our Live Screen, we will add the machine name (port name) to the front of each machine and behind it the current state of the machine. Insert the following Real-time Values at the top of the list.

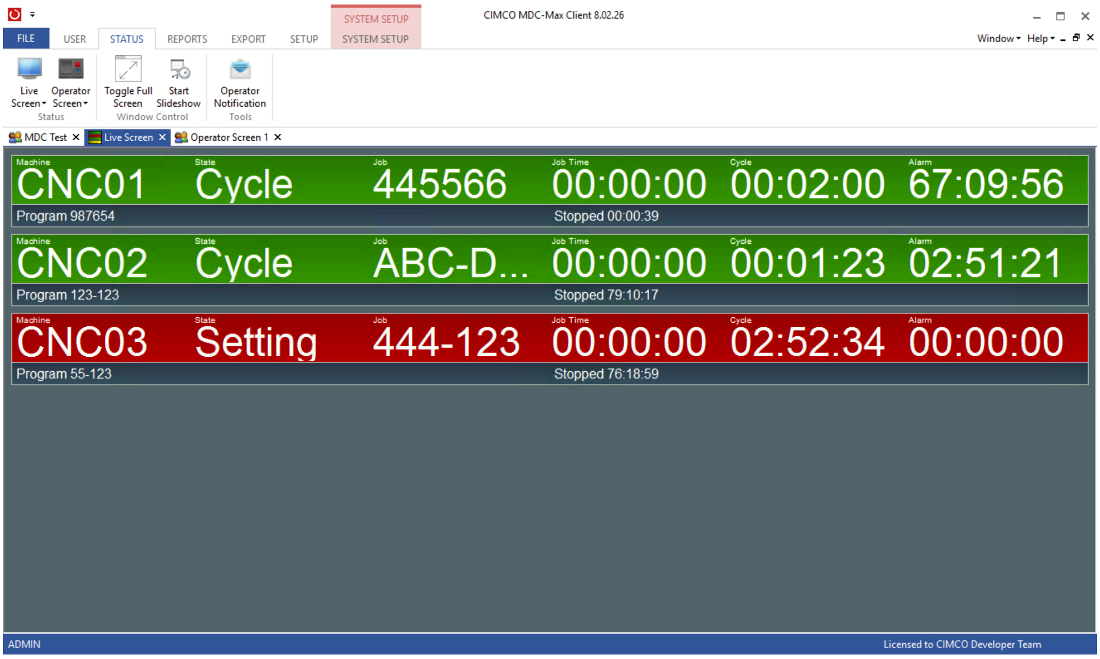
Value (expression)	**Title**	**Format**
TEXT[PORTNAME]	Machine	TEXT
TEXT[STATE]	State	TEXT

3. If a downtime reason has been selected/triggered using our Operator Screen 1, it would also be useful to show this information instead of the Alarm state so that the Live Screen writes e.g. Cooling Fluid instead of just Alarm.

   To do this, select **States** in the treeview menu under **Status Screens** and add the following additional states. Make sure they have a higher priority than the existing.

Value (expression)	**Description**	**Color**
TIMERRUNS[SETTING]	Setting	Red
TIMERRUNS[TOOLING]	Tooling	Red
TIMERRUNS[COOLINGFLUID]	Cooling Fluid	Red
TIMERRUNS[CHIPS]	Chips	Red
TIMERRUNS[OKTORUN]	Ok To Run	Red

4. Click the **Save/Close** button in the top menu and see how the Live Screen now displays the machine name (port name) first and then behind it the current state. Also notice that when a downtime reason is clicked in Operator Screen 1 the downtime reason is also shown on the Live Screen.



Machine	State	Job	Job Time	Cycle	Alarm
CNC01	Cycle	445566	00:00:00	00:02:00	67:09:56
Program 987654			Stopped 00:00:39		
CNC02	Cycle	ABC-D...	00:00:00	00:01:23	02:51:21
Program 123-123			Stopped 79:10:17		
CNC03	Setting	444-123	00:00:00	02:52:34	00:00:00
Program 55-123			Stopped 76:18:59		

5. We also need to add the downtime reasons/states to our Timeline. Go to **System** > **System Setup** > **Tables Charts** and select **Log / Timeline** in the treeview menu.

   We already have three events / Timeline triggers included in our timeline which are triggered whenever their Timers start running. To include the downtime reasons, add them in the following way and make sure they have a higher priority than the existing.

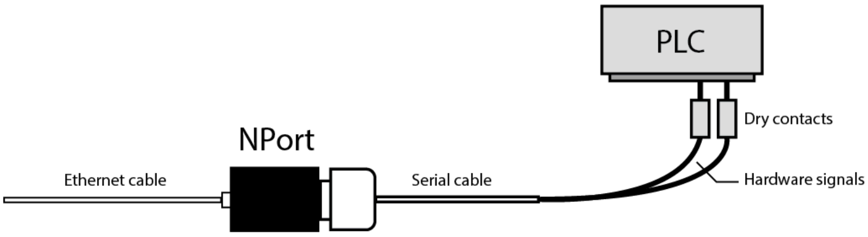
Event/Timeline trigger	**Title**	**Color**
TIMERRUNS[SETTING]	Setting	Red
TIMERRUNS[TOOLING]	Tooling	Red
TIMERRUNS[COOLINGFLUID]	Cooling Fluid	Red
TIMERRUNS[CHIPS]	Chips	Red
TIMERRUNS[OKTORUN]	Ok To Run	Red

6. Click the **Save/Close** button in the top menu and open the **Timeline**. Use the Operator Screens to set machine state and downtime reasons and see how this is reflected in the Timeline. In the timeline, the current state is shown in the tooltip on mouse-over.

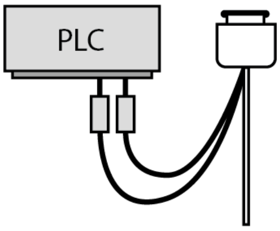
11. Connection using MOXA NPort 5110 (simplified)

To establish a connection to machines that are not equipped with a data interface, one of the simplest options is to use a MOXA NPort device server.

On one side the NPort has a COM port for a serial RS-232 cable that must be wired to the machine's PLC relays, and on the other it has an ethernet port for connecting to the local network or PC. That way, a connection from the machine to the DNC-Max Server can be established.

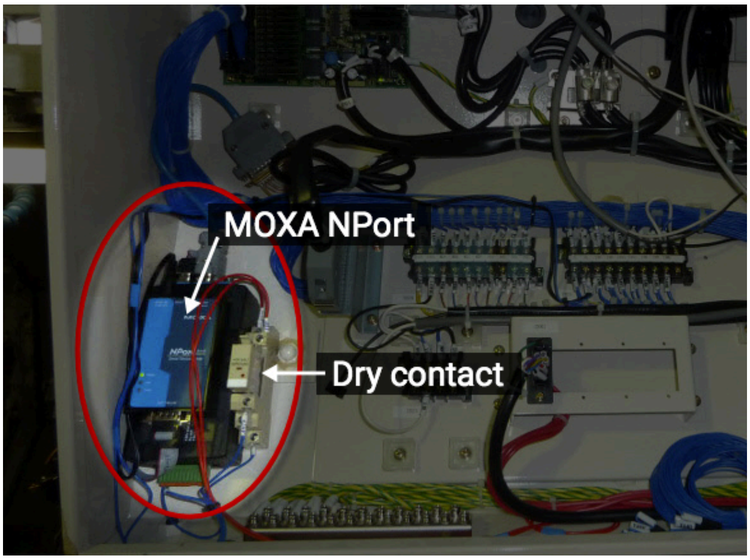


If the machine is equipped with a COM port for data transmission, an existing serial cable on that port can also be used for transmission of hardware signals since only a few of the pins/wires in the cable are dedicated to data transmission. The rest can be wired to the machine's PLC relays to "pull" the hardware signals required by the MDC system.



A machine's PLC will (by standard) output 24volt. The COM port on the MOXA NPort can, however, only handle +/- 12volt. Connecting the serial cable to the NPort will therefore damage the NPort. To avoid this, we recommend installing a so-called "dry contact", which enables a contactless connection from the PLC relays to the cable wires. That way, the machine's electronics are completely separated from the NPort.

In most cases, customers will have their own electricians who can easily wire the machines, so the hardware signals that are required by the MDC system can be collected. It is always preferable to let customers make the changes themselves, so you are not responsible if the machines get damaged.



11.1. Testing input (at the office)

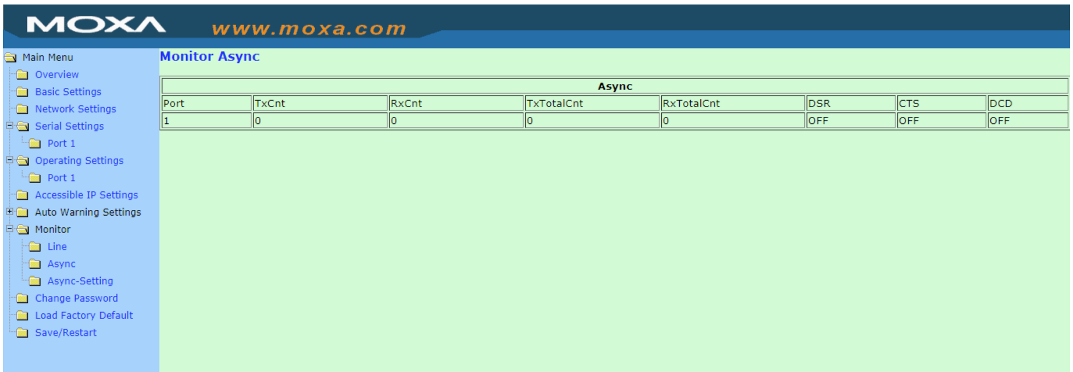
While still at your office, before going to the customer, it is a good idea to make sure the device is working and that the signals are passing through to the DNC-Max Server. Follow these steps to check that the Nport and connection to the machine or test device is working:

1. Unbox the Nport and check the back of the device. You will notice that the device has a factory set IP address such as 192.168.127.254. All Nports are "born" with the same factory IP.

2. Use an Ethernet cable to connect the device to your PC.

3. In Windows, set a static IP address for your PC. Make sure the static IP is different from that of the Nport. Refer to Google on how to do this.

4. Open a browser and enter the IP address of the Nport in the address bar/url. You will now have access to the Nport's configuration interface.



Monitor Async

Async

Port	TxCnt	RxCnt	TxTotalCnt	RxTotalCnt	DSR	CTS	DCD
1	0	0	0	0	OFF	OFF	OFF

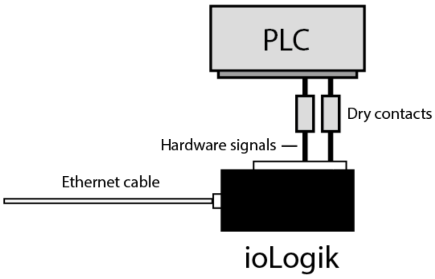
5. To check the status of the COM port, open **Monitor / Async**. Here you can see the DSR, CTS, and DCD. When signals are received on these pins, their status will change to ON.

At the point where the Nport has to be installed permanently as part of the customer's network, its IP address will have to be changed to eiher DHCP or a customer-defined IP address. Also, if connecting multiple Nports, you will have to change their factory IP address, so they do not use the same address.

12. Connection using MOXA ioLogik E1210 (simplified)

To establish a connection to machines that are not equipped with a data interface, an alternative to the Nport and a more professional option is to use a MOXA ioLogik E1210 device server.

The ioLogik features 16 inputs where wires from the PLC relays can be connected. That way, you can avoid making changes to any existing serial connection which might be used by other systems. When using an ioLogik, the installation looks clean and professional.



We recommend installing dry contacts which enables a contactless connection from the PLC relays to the cable wires. That way, the machine's electronics are completely separated from the ioLogik.

In most cases, customers will have their own electricians who can easily wire the machines, so the hardware signals that are required by the MDC system can be collected. It is always preferable to let customers make the changes themselves, so you are not responsible if the machines get damaged.

12.1. Testing inputs (at the office)

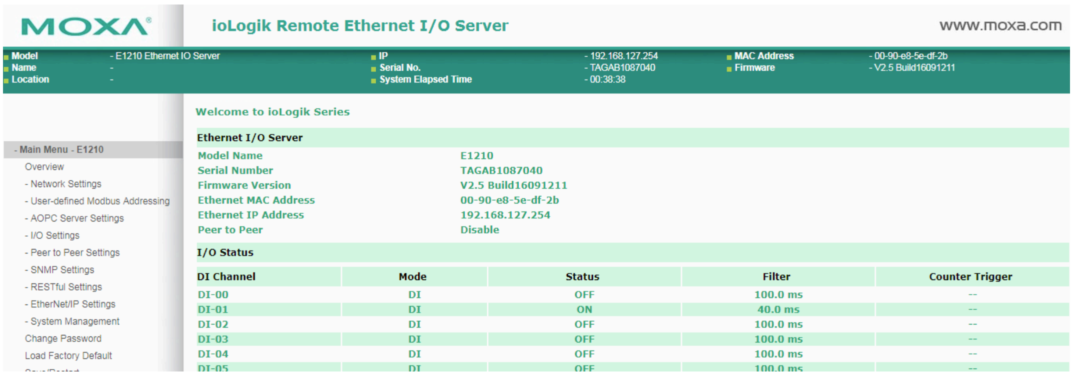
Follow these steps to check that the ioLogic and connection to the machine or test device is working. This can be done at the office, before going to the customer, to make sure the device is working and that the signals are passing through to the DNC-Max Server.

6. Unbox the ioLogik and check the back of the device. You will notice that the device has a factory set IP address such as 192.168.127.254. All ioLogik devices are "born" with the same factory IP.

7. Use an Ethernet cable to connect the device to your PC.

8. In Windows, set a static IP address for your PC. Make sure the static IP is different from that of the ioLogik. Refer to Google on how to do this.

9. Open a browser and enter the IP address of the ioLogik in the address bar/url. You will now have access to the ioLogik's configuration interface.



Model - E1210 Ethernet IO Server
Name -
Location -
IP - 192.168.127.254
Serial No. - TAGAB1087040
System Elapsed Time - 00:38:38
MAC Address - 00-90-e8-5e-df-2b
Firmware - V2.5 Build16091211

Welcome to ioLogik Series

Ethernet I/O Server

Model Name	E1210
Serial Number	TAGAB1087040
Firmware Version	V2.5 Build16091211
Ethernet MAC Address	00-90-e8-5e-df-2b
Ethernet IP Address	192.168.127.254
Peer to Peer	Disable

I/O Status

DI Channel	Mode	Status	Filter	Counter Trigger
DI-00	DI	OFF	100.0 ms	--
DI-01	DI	ON	40.0 ms	--
DI-02	DI	OFF	100.0 ms	--
DI-03	DI	OFF	100.0 ms	--
DI-04	DI	OFF	100.0 ms	--
DI-05	DI	OFF	100.0 ms	--

10. Each of the DI Channels corresponds to an input on the ioLogik and their status are currently set to OFF since no wires have been connected.

11. Once you connect a wire, you will see the status change to ON indicating that the input is indeed working.

At the point where the ioLogik has to be installed permanently as part of the customer's network, its IP address will have to be changed to eiher DHCP or a customer-defined IP address. Also, if connecting multiple ioLogik devices, you will have to change their factory IP address, so they do not use the same address.

13. Plan the demo

When setting up an MDC demo at a customer's site there are a few things to keep in mind.

Messages / hardware signals
In order to provide a useful demo, agree with the customer in advance which hardware signals should be collected from the machines. Also, the hardware signals determine how the connection / wiring of the machine should be done.

Machine(s) and wiring
It is up to you and the customer how many machines will be connected as part of the demo. Initially, it is usually only one or a few. Find out what is required to connect the machines to the demo PC. Let the customer's own technician/electrician be responsible for providing connections to the machine's hardware signals - especially if any wiring is required. All this should of course be prepared in advance.

Demo PC
If you bring your own PC, you can set it up beforehand and make sure everything in the MDC system is working. If using a customer PC, make sure the customer sets it up (beforehand) for you to have a Windows account with administrator priviledges. It is possible to confgure the MDC system beforehand (at your office) and transfer the configuration to their PC.

Device server
If the setup requires device servers such as the MOXA Nport 5110 or the MOXA ioLogik E1210, it is a good idea to configure these at the office and check that they are working before going to the customer.

Make a plan
In short, make a project plan. Find out what responsibilities you and the customer have. Prepare as much as possible in advance so that when you arrive to set up the demo, you can do it in a professional and efficient manner.

In many organizations, a simple thing such as getting administrator privileges in Windows can take a long time. The same goes for accessing the local network and connecting machines. If those things have not been planned in advance and taken care of, you can easily end up wasting a day – not to mention that you will also waste the customer's time, which does not leave a great impression.

From experience, once a demo is set up and the customer can see how his own data is collected and reflected in Live Screens, Timelines, and reports, only then does he realize the potential and usefulness in regard to his own operation.